

FIBER OPTIC LINEAR
HEAT DETECTION
(FOLHD)

Volume II

Operations & Maintenance Manual

This Page Left Intentionally Blank

LIOS FIBER OPTIC LINEAR **HEAT DETECTORS (FOLHD)**

Operations & Maintenance Manual
December 2015

Fibre Optic Linear Heat Detection Optimum Safety in Tunnels and Special Hazard Buildings



DE.TECT® LHD3 Linear Heat Detector for Fire Detection

LIOS DE.TECT is the state of the art fire detection system to provide optimum safety for fire detection in tunnels and other special hazard areas due to its ability to cope with extreme environmental conditions.

Rising demand for fire surveillance in facilities like rail and service tunnels, cable routes and ducts, storage facilities and warehouses or conveyor belts and many other special hazard facilities motivated LIOS Technology to offer a measuring system tailor-made for these industrial applications.

A single evaluation unit of the DE.TECT series provides fast and spatially well-resolved fire detection along optical fibres with ranges of up to 10km per channel.

www.de-tect.pro

Technical Highlights

- Linear heat detector for quick fire detection and precise localization of the fire source
- Alarm and pre-alarm for each zone
- Selectable alarm and pre-alarm criteria
- High spatial resolution – down to 0.25 m
- Signal processing based on OFDR-Technology (Optical Frequency Domain Reflectometry)
- Available with 1, 2 and 4 channels
- 1000 free programmable zones for each channel
- Information regarding magnitude and the direction of the fire spread
- Redundant sensor system applicable
- Suitable for wind speeds of up to 10 m/s
- Laser product class 1M according DIN EN 60825-1: 2007
- VdS approval prEN 54-22 G 211030
- UL approval

Road and Railway Tunnels ■ Conveyor Belts ■ Mining ■ Service Tunnels ■ Cable Ducts ■ Storage Facilities ■ Warehouses ■ and many other Special Hazard



DE.TECT LHD models – LINEAR HEAT DETECTION

1 channel models	1, 2, 4, 6, or 10 km sensor cable (distance range)
2 channel models	1, 2, 4, 6, or 10 km sensor cable per channel
4 channel models	1, 2, 4 or 6 km sensor cable per channel

Mechanical data

Controller	19" Rack / 3 units of height
Dimensions (H x W x D)	13.5 x 44.9 x 29 cm
Colour	grey
Weight	13kg

Electrical data

Operating voltage (DC Controller)	DC 12 ... 48 V
Mains voltage (AC Controller)	AC 100 ... 240 V
Power consumption (DC Controller)	<25W (max. 45 W/60°C)
Programmable inputs	4 (optional up to 40)
Programmable outputs (potential-free)	12 (optional up to 106)
Communication interfaces	LON, TCP/IP, Modbus, RS232, USB

Optical data

Fibre Type	Gradient index 62.5/125 µm multimode
Optical connector	E2000 / APC
Laser classification	Class 1M (EN60825-1: 2007)

Environmental conditions

Storage temperature	-35 ... +75 °C
Operating temperature	-10 ... +60 °C
Humidity (non condensing)	≤95 % rel.
Protection class (IEC 60529)	IP51

Approvals

VdS (pr EN 54-22)	G 211030, response classes A1N, BN, CN
UL	S25135



The New Standard in Modern Fire Detection

© 2014 Copyright by LIOS Technology GmbH
 Data and design subject to change without notice.
 Supply subject to availability.
 LIOS Technology and DE.TECT are registered trademarks.
 Document: LIOS DE.TECT Datasheet Edition: 18.07.2014

LIOS Technology GmbH – Linear Optical Sensors
 Schanzenstrasse 39 / Building D9-D13
 51063 Cologne, Federal Republic of Germany
 Phone +49 221 99887-0 / Fax +49 221 99887-150
 info@lios-tech.com / www.lios-tech.com

This Page Left Intentionally Blank

INSTALLATION MANUAL

OTS3- / LHD3-Series Controllers

Introduction

Safety Instructions

Installation

Maintenance and Repair

Technical Data

Information Scheme

Contact Information



Table of Contents

- Table of Contents 2
- 1 Safety Instructions..... 5
 - 1.1 Safety Guidelines..... 5
 - 1.2 General Safety 5
 - 1.3 Electrical Safety 5
 - 1.4 Laser Safety 6
 - 1.5 Application-Specific Safety 7
 - 1.5.1 Fire Detection 7
 - 1.5.2 Railway Tunnels..... 7
 - 1.5.3 Power Cable Monitoring..... 8
 - 1.5.4 Explosive Areas..... 8
- 2 Introduction..... 9
 - 2.1 General Information..... 9
 - 2.1.1 Purpose of the Installation Manual..... 9
 - 2.1.2 Legal Notices..... 9
 - 2.1.3 Qualified Personnel 9
 - 2.1.4 Controller Model and Range 9
 - 2.2 System Architecture 11
 - 2.3 Function..... 11
 - 2.4 Controls, Interfaces, Indicators and Buzzer..... 12
 - 2.4.1 Front Panel of OTS3-Series Controllers 12
 - 2.4.2 Front Panel of LHD3-Series Controllers..... 13
 - 2.4.3 Rear Panels 14
 - 2.5 Intended Use 16
 - 2.6 Applications..... 17
 - 2.6.1 Fire Detection 17
 - 2.6.2 Power Cable Monitoring..... 17
 - 2.6.3 Downhole Monitoring 18
 - 2.6.4 Industrial Process Monitoring 18
 - 2.6.5 Leak Detection at Dams, Pipelines and Tanks 18
 - 2.6.6 Environmental Monitoring 19
 - 2.6.7 Radioactive Areas 19
 - 2.6.8 Explosive Areas..... 19
- 3 Installation..... 20



- 3.1 Hardware Installation 20
 - 3.1.1 Unpacking the Controller 20
 - 3.1.2 System Components and Tools 21
 - 3.1.3 Controller Location 22
 - 3.1.4 Rack Mounting..... 22
 - 3.1.5 Connecting Electrical Cables..... 23
 - 3.1.6 Connecting Sensor Cable(s) 35
- 3.2 Sensor Cable Installation 39
 - 3.2.1 Cable Selection 39
 - 3.2.2 Fibre and Sensor Cable Recommendations..... 40
 - 3.2.3 Handling of Sensor Cables 41
 - 3.2.4 Laying of Sensor Cables 42
 - 3.2.5 Laying Plan..... 42
 - 3.2.6 Laying Plans for Fire Detection 42
 - 3.2.7 Connecting Sensor Cables 43
 - 3.2.8 Termination of Sensor Cables..... 46
 - 3.2.9 Repair of Sensor Cables 47
 - 3.2.10 Ceiling Mounting 48
 - 3.2.11 Fibre-in-Metal-Tube for Power Cables 50
- 4 Maintenance and Repair 51
 - 4.1 Maintenance..... 51
 - 4.1.1 Cleaning of Internal Pigtails..... 51
 - 4.2 Repair 52
 - 4.2.1 Exchange of Electrical Fuse 53
 - 4.2.2 Exchange of Memory Card 53
- 5 Technical Data 54
 - 5.1 General Technical Data 54
 - 5.2 Transport and Storage Conditions..... 55
 - 5.3 Interfaces..... 55
 - 5.4 Standards and Approvals..... 57

The LHD3 series controllers are approved by UL/ULC and they have been evaluated for open-area protection spacings as outlined in the table 24. 58

 - 5.5 Hazardous Substances and Recycling 58
- 6 Information Scheme and Contact Information 58
 - 6.1 Information Scheme 58



- 6.2 Contact Information 59
 - 6.2.1 Online Availability..... 59
 - 6.2.2 Representatives and Offices..... 59
 - 6.2.3 Training..... 59
 - 6.2.4 Technical Support 59
 - 6.2.5 Contact Information 59
- Figures 61
- Tables 63
- Abbreviations 64
- Glossary 65







1 Safety Instructions

1.1 Safety Guidelines

This manual contains notices you have to consider in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol. Notices referring only to property damage have no safety alert symbol. The notices shown in Table 1 are graded according to the degree and cause of danger.

Table 1: Safety symbols and notices

	Warning!	This symbol and label indicate that death or severe personal injury may result if proper precautions are not taken.
	Warning!	This symbol and label indicate that death or severe personal injury may result from electricity if proper precautions are not taken.
	Warning!	This symbol and label indicate that severe personal injury may result from laser radiation if proper precautions are not taken.
	Warning!	This symbol and label indicate that death or severe personal injury may result from explosion if proper precautions are not taken.
	Caution!	This label without a safety alert symbol indicates that property damage can result if proper precautions are not taken.

1.2 General Safety

Caution!

The controller shall only be used as intended by the manufacturer (see section 2.5).

Switch the controller off before connecting or disconnecting any optical fibre sensor cable. Never pull-off an optical fibre connector from an E2000 optical port during an active measurement. The emitted laser radiation may damage the fibre endfaces of pigtail and controller.

Never connect pigtails with contaminated ferrule endfaces to the controller. Contaminated endfaces may easily be damaged by laser radiation (see section 3.1.6.4).

1.3 Electrical Safety



Electricity can be a danger to health and life. See a doctor immediately after contact with dangerous voltages.

The controller contains components carrying dangerous voltages. Do not open or repair the controller. Do not operate the controller with damaged housing.

Verify that the supply voltage is within the range given at the DC or AC supply socket of the controller before connecting the power supply cable.

The AC power supply cable has to be disconnected before replacing the fuse at the power supply of the controller.



Accumulator terminals may never be shorted or come into contact with the controller housing.

National safety regulations for using electricity must be considered. Electricity supply and earthing must be carried out in accordance with national regulations.

1.4 Laser Safety



Warning!

The controller contains a laser source. Caution, the laser radiation is invisible! The controller is classified as laser class 1M product. The following label is attached close to the fibre connector(s) on the controller:



The laser emits continuously modulated laser radiation at a wavelength of 1550 nm. The modulation frequency is between 500 Hz and 500 MHz. The maximum average power is 110 mW, and the maximum peak power is 200 mW.

The accessible laser radiation represents no danger for the human eye or skin assuming that:

- The radiation is not focussed by optical instruments like magnifying glasses, lenses or microscopes and
- The end of an optical fibre or cable is not approached closer than 10 cm to the eye or skin.

With optical instruments or during direct approach of the optical fibre to the human body, laser radiation can generate heat that damages the skin or the cornea of the eye. Such damages are wet bulges, cataract and burnings of cornea and skin. See a doctor immediately in case of eye damages.

National regulations on laser safety have to be considered during installation and operation of the controller.

The controller must be switched off before directly viewing on fibre ends, splices or optical fibre connectors with or without optical instruments. In addition, the warning label shown in Figure 1 has to be placed close to the mains switch of the controller. Use only CE marked fibre scopes equipped with a 1550 nm DIR LB3 laser safety filter according to EN 207.

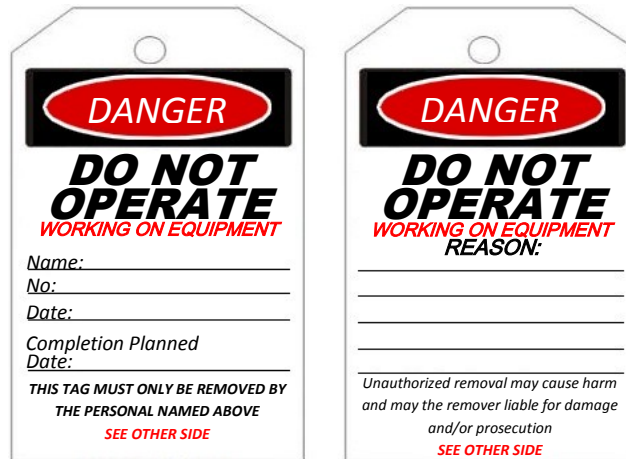


Figure 1: Warning label

Optical fibre ends must not be directly approached to the eye or the skin. Laser radiation emitted from fibre ends must not be focussed with optical instruments onto the eye or the skin.

The controller must be operated with attached fibre(s). Unused E2000 optical fibre ports of the controller must be terminated using dummy connectors.

Emission of laser radiation at the end of a sensor fibre or cable must be prevented by using an optical termination box or a termination kit (see section 3.2.8).

The laser is automatically switched on by starting a measurement. The LEDs designated with “LASER ACTIVE / LASER ON” at the front and rear panels of the controller indicate laser operation.

If an optical fibre sensor cable, pigtail or optical fibre is broken, it must be repaired, terminated or replaced immediately (follow the instructions given in section 3.2.9).

1.5 Application-Specific Safety

1.5.1 Fire Detection

Caution!

During the commissioning and testing of fire detection controllers (LHD3-series) it must be ensured that:

- false triggering of transmission devices is securely prevented,
- the fire brigade is informed before test triggering of transmission devices,
- triggering of fire extinguishing controls is blocked,
- the responsible person is present when third party devices are triggered,
- the collective fault output of the controller is connected to the alarm control system,
- the collective alarm output of the controller is connected to the alarm control system, and
- a programmable output is assigned to the test mode and connected to the alarm control system if the test mode shall be used.

1.5.2 Railway Tunnels



Warning!



A safe distance from high voltage lines must be maintained in railway tunnels with electrical traction lines. The safety distance depends on the maximum operating voltage and national or local regulations. Safety distances must be confirmed and signed by the tunnel operator.

Only splice boxes with grounding may be used for metal containing optical fibre sensor cables. Alternatively, metal-free sensor cables may be used without grounding.

1.5.3 Power Cable Monitoring



Warning!

Consider the danger of electrical voltage when working in the vicinity of power cables. They may carry life threatening voltage levels.

Only splice boxes with grounding may be used for metal containing optical fibre sensor cables. Alternatively, metal-free sensor cables may be used without grounding.

The grounding of metal containing optical fibre sensor cables must take into account induced currents and has to be compliant with the overall grounding concept of the power cable installation.

1.5.4 Explosive Areas



Warning!

The sensor cable must be operated with an approved OTS3/LHD3-EX-series controller model. The OTS3/LHD3-EX-series controller must be installed and operated outside the explosive area or inside an accordingly approved and marked EX –d protection box. Details on the restrictions of using the OTS3/LHD3-EX-series controllers, protection boxes and the sensor cables in explosive areas are given in the approval documents. The OTS3/LHD3-EX-series controllers are approved according to the European directive 94/9/EC. The compliance to UL/ULC standards is not evaluated.

The OTS3/LHD3-EX-series controller must be retested by LIOS Technology every 10 years in order to confirm the compliance to the explosion safety standards EN 60079-0 and -28.

Only sensor cables that are no source of ignition and do not cause other danger of explosion, e.g. by electrostatic discharge, may be used in explosive areas. Always ensure that no danger emerges from mechanical damage to the sensor cable. This applies to all production, installation, maintenance work and to regular operation in the object being monitored.

Conductive sensor cables must be protected against the ingress of energy from other sources (current sources, lightning, electrostatics, electromagnetic or magnetic fields) and its transmission into the explosive area. This requires a proper grounding, generally just outside the explosive area.

Splicing of fibres is forbidden in explosive areas. The use of spliced fibre connections in these areas is not intended by LIOS Technology. LIOS Technology does not offer any splice boxes for that purpose. If the operator of a facility wants to use splice boxes, accordingly approved splice boxes have to be chosen. LIOS Technology does not assume any liability for the use of splice boxes in explosive areas.

National regulations must be considered during work in explosive areas. The installer and operator of the facility are responsible for safe installation, not the manufacturer of the controller.



2 Introduction

2.1 General Information

2.1.1 Purpose of the Installation Manual

This Installation Manual provides information based on the requirements for engineering documentation. The information relates to the OTS3- and LHD3-series controllers, its place of use, transportation, storage, installation, commissioning and maintenance. These instructions are intended for:

- Commissioning engineers
- Service technicians
- Maintenance technicians

General knowledge of electrical engineering, optical fibres, electrical safety and laser safety is needed to understand the installation and operating instructions. It is also assumed that those using the Installation Manual have experience in using personal computers with Windows operating systems.

Please read chapter 1 “Safety Instructions” carefully.

2.1.2 Legal Notices

We have reviewed the contents of this publication to ensure consistency with the hardware and software described herein. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

It is our policy to constantly improve the design and specifications. Accordingly, the details represented in this manual cannot be regarded as final and binding.

The contents of this manual, in particular all texts, pictures, graphics, and logos are subject to the copyright of LIOS Technology GmbH. Any printing, copying, publishing, presentation, translation or modification of the manual or parts of its contents requires a written permission from LIOS Technology GmbH.

2.1.3 Qualified Personnel

The controller may only be installed and used in conjunction with this documentation. Installation, commissioning and operation of the controller may only be performed by qualified personnel. Within the context of the safety instructions in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

2.1.4 Controller Model and Range

The controller model is identified on the nameplate on the rear panel of the controller (Figure 2).

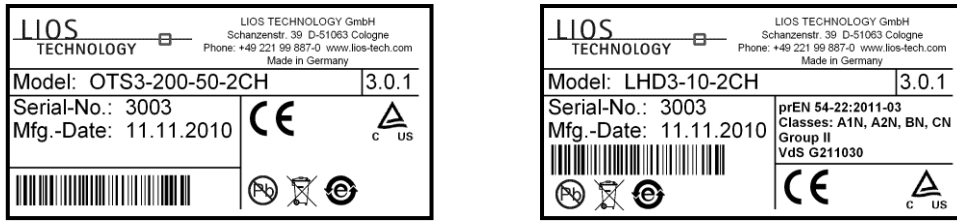


Figure 2: Examples of the nameplates of OTS3-series and LHD3-series controllers

Controller models are

OTS3-xx0-yy(-EXTn)(-zzCH) or

LHD3-xx(-EXTn)(-zzCH)

using the assignments given in Table 2.

Table 2: Controller models

<i>Controller series and model code</i>	<i>Assignment</i>
OTS3	Optical temperature sensing controller for distributed temperature sensing
LHD3	Linear heat detection controller for fire detection
xx	Distance range per channel in km
yy	Fibre core diameter
EXTn (option)	Controller for monitoring explosive areas of temperature class Tn (n=3 or 4)
zz (option)	Number of optical fibre channels

Standard models are equipped with a DC power supply. An AC power supply is optionally available. The AC power supply may also be operated using DC voltages within a certain range (Table 19). The use of an AC power supply in fire detection was not evaluated by UL/ULC or VdS.



2.2 System Architecture

Figure 3 shows the basic architecture of an optical fibre sensing system comprising the controller, sensor cables and optional external devices.

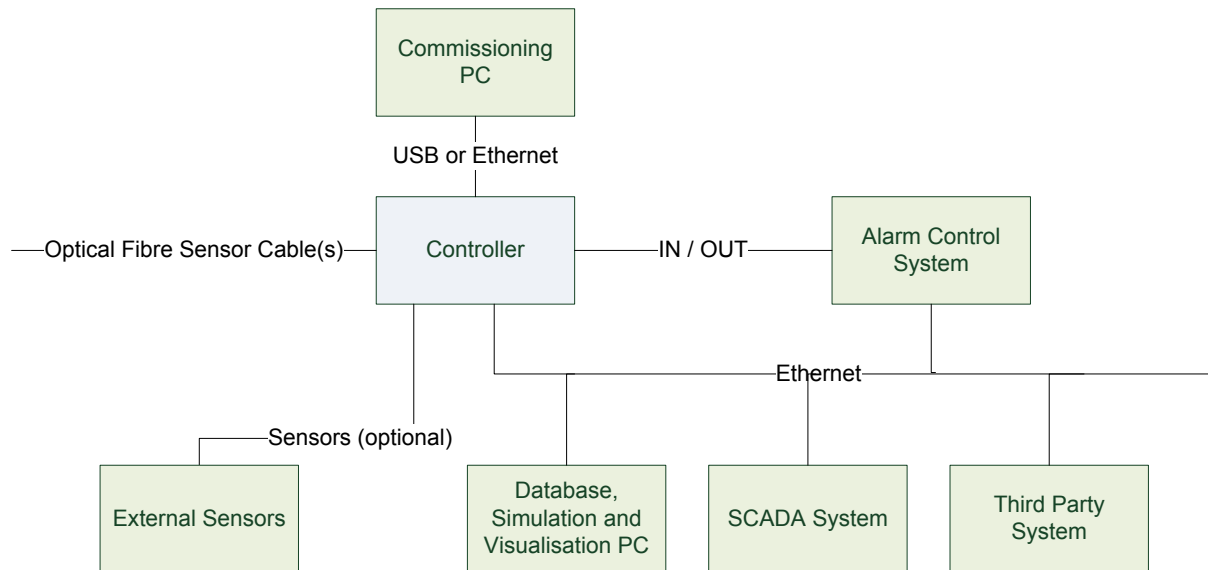


Figure 3: System architecture

The controller may be connected to other system components as follows:

- Optical fibre sensor cable(s) using E2000/APC optical fibre connector(s) marked with “Ch. xx”
- Computer(s) for installation, commissioning, data storage, simulation and visualization using the USB or Ethernet interfaces
- An alarm control system using the input and output relays and/or the Ethernet interface
- SCADA systems and other third party systems using the Ethernet interface
- External sensors for reference temperatures, currents and other physical quantities using the Ethernet or the optional Sensors interface
- DC or AC power supply using the respective power supply socket.

2.3 Function

The controller provides spatially resolved temperature profiles along optical fibre sensor cables. The sampling resolution and measurement time may be set by the user within the specified ranges. Longer measuring times result in a better temperature resolution. The longer the sensor cable and the smaller the sampling resolution, the longer measuring times are required to achieve a certain temperature resolution.

The controller software also evaluates temperature profiles as well as their temporal and local variations according to maximum or minimum temperature, slope and local differential criteria in order to generate fire alarms or pre-alarms.

The controller is intended for stand-alone operation without a permanent link to a computer. Computers may be connected for commissioning, control and data download.

The controller comprises:

- An optical system including a laser transmitter and an optical receiver



- Electrical signal amplifiers and filters
- A digital signal processing and control system including a LINUX based communication unit
- USB, Ethernet, EIA-232 (RS232), relay and optional sensor interfaces
- LED indicators for operation state, alarms and errors
- A DC or optional AC power supply
- A fibre switch (option)
- An enclosure serving as passive heat sink

The controller does not contain any moving mechanical parts except for the optional fibre switch comprising a MEMS switch.

2.4 Controls, Interfaces, Indicators and Buzzer

2.4.1 Front Panel of OTS3-Series Controllers

Figure 4 shows the front panel of an OTS3-series controller. There are three groups of LEDs on the front panel indicating operation, communication and alarms of the controller as given in Table 3.

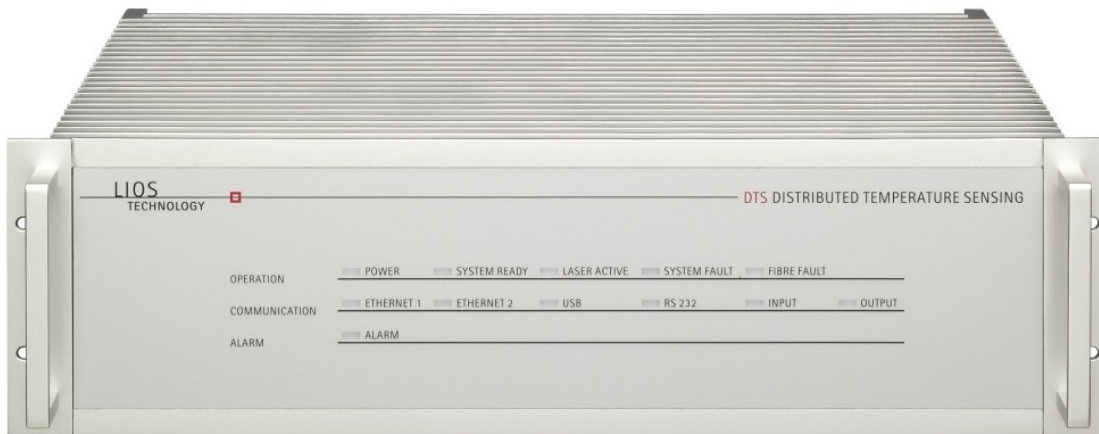


Figure 4: Front view of an OTS3-series controller

Table 3: Front LEDs of OTS3-series a controller

Group	LED Designation	Colour	Function
Operation	POWER	Green	Power supply is operating. This LED is controlled by the digital system. It is therefore time-delayed to the POWER ON LED on the rear panel by a few seconds.
	SYSTEM READY	Green	Controller is ready.
	LASER ACTIVE	Green	Laser emission is enabled.
	SYSTEM FAULT	Yellow	System fault. Check error messages for details (see Operation Manual OTS3).
	FIBRE FAULT	Yellow	Fibre fault (e.g. fibre break). Check sensor cable(s) (see Operation Manual OTS3).
Communication	ETHERNET 1	Green	Ethernet 1 link is present. Blinking indicates data transmission.
	ETHERNET 2	Green	Ethernet 2 link is present. Blinking indicates data transmission.
	USB	Green	Data transmission over USB.
	RS232	Green	Data transmission over RS232.
	INPUT	Green	Input(s) are active.



	OUTPUT	Green	Output(s) are active.
Alarm	ALARM	Red	Alarm(s) are active. (A blinking LED indicates that a previous alarm has not yet been acknowledged.)

2.4.2 Front Panel of LHD3-Series Controllers

Figure 5 shows the front panel of a LHD3-series controller. There are three groups of LEDs on the front panel indicating operation, communication and channel status of the controller as given in Table 4. Three channel status LEDs are provided for each of the up to four channels.

An ACKNOWLEDGEMENT key switch for acknowledging alarms and resetting of the controller is located on the right of the front panel. The switch controls the actions given in Table 5. Depending on the controller configuration, actuation of the key is also confirmed over interfaces and/or outputs (see Operation Manual).



Figure 5: Front view of a 4-channel LHD3-series controller

Table 4: Front panel LEDs of a LHD3-series controller

Group	LED Designation	Colour	Function
Operation	POWER	Green	Power supply is operating. This LED is controlled by the digital system. It is therefore time-delayed to the POWER ON LED on the rear panel by a few seconds.
	SYSTEM READY	Green	Controller is ready.
	LASER ON	Green	Laser emission is enabled.
	SYSTEM FAULT	Yellow	System fault. Check error messages for details (see Operation Manual LHD3).
Communication	ETHERNET 1	Green	Ethernet 1 link is present. Blinking indicates data transmission.
	ETHERNET 2	Green	Ethernet 2 link is present. Blinking indicates data transmission.
	USB	Green	Data transmission over USB.
	RS232	Green	Data transmission over RS232.
	INPUT	Green	Input(s) are active.
	OUTPUT	Green	Output(s) are active.
Channel status	FIBRE FAULT	Yellow	Fibre fault (fibre break). Check fibre(s) (see Operation Manual LHD3).
	PRE-ALARM	Red	Pre-alarm(s) are active. (A blinking LED indicates that a previous pre-alarm has not yet been reset.)



ALARM	Red	Alarm(s) are active. (A blinking LED indicates that a previous alarm has not yet been reset.)
-------	-----	---

Table 5: Functions of the ACKNOWLEDGEMENT key switch

Duration of Actuation	Function	Action
Short (< 3 s)	Acknowledgement	All alarms and pre-alarms are acknowledged. The buzzer is deactivated.
Long (> 3 s)	Reset	All alarms, pre-alarms, faults and manual reset outputs are reset if they are no longer active. The corresponding alarm, pre-alarm and fibre or system fault LEDs are deactivated. Self-reset outputs are also reset if this function is activated (see Operation Manual).

2.4.3 Rear Panels

The rear side of the controller consists of three panels for optical ports, electrical interfaces and power supply (Figure 6). Two green LEDs indicate operation of the power supply and the laser (see Table 6). In dependence on the controller model and the options, there are different versions of each panel.

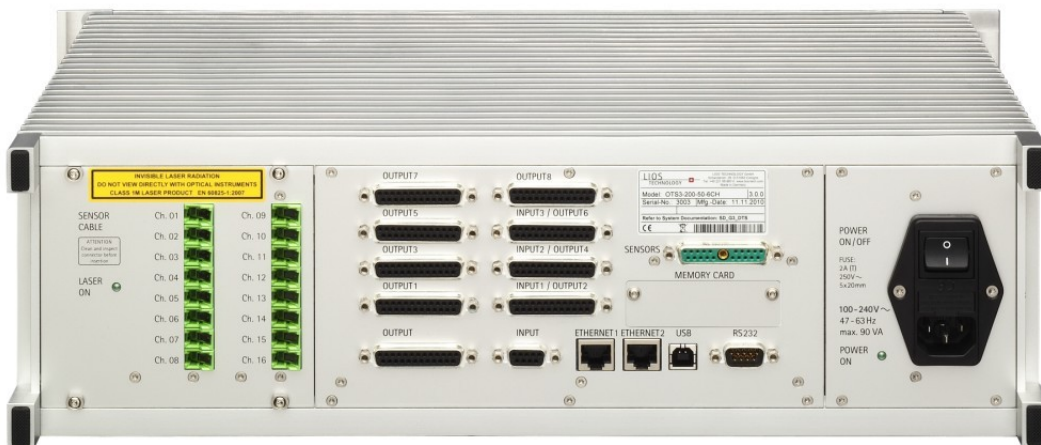


Figure 6: Rear view of a controller with optical ports, electrical interfaces and power supply panels

Table 6: Rear panel LEDs of the controller

Panel	LED	Colour	Function
Optical ports	LASER ON	Green	Laser operation is enabled.
Power supply	POWER ON	Green	Power supply is operating.

2.4.3.1 Optical Ports Panel

The optical ports panel comprises the LASER ON LED, the laser warning label and a number of optical ports corresponding to the controller model. Typical optical ports panels are shown in Figure 7. Each optical port is marked with the channel number.



Figure 7: Optical ports panels of 2, 8 and 16 channel controllers

2.4.3.2 Interfaces Panel

In dependence on the controller model, a standard or an extended interface panel is provided. The standard interface panel (Figure 8) comprises the interface sockets (Output, Input, Ethernet 1, Ethernet 2, USB and RS232), a memory card slot and the controller nameplate (Figure 2). The extended interface panel (Figure 9) in addition comprises optional sockets for extra inputs and outputs as well as for external sensors.

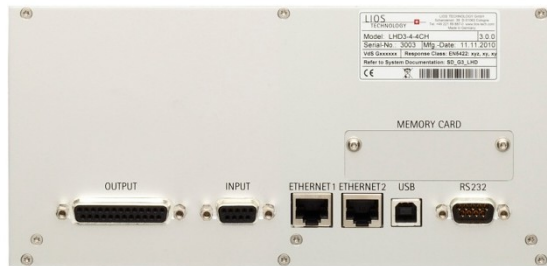


Figure 8: Standard interface panel

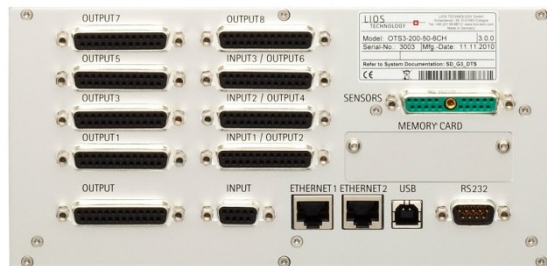


Figure 9: Extended interface panel

2.4.3.3 Power Supply Panel

The power supply panel comprises the POWER switch, the POWER ON LED, the fuse and the AC or DC supply socket (Figure 10).



Figure 10: DC (left) and AC (right) power supply panels

2.4.3.4 Buzzer

The controller comprises an internal buzzer indicating start-up, faults and alarms using codes given in Table 7. The activity of the buzzer for faults and alarms may be set using the software Charon3 (see Operation Manual).

Table 7: Buzzer beep code for different events

Event	Mode	Duration (Pattern)
Power on	Long single beep	Few seconds.
Ping (test of IP connection)	Fast periodic beeping	3 s (0.07 s on / 0.03 s off)
Fault (not acknowledged)	Slow periodic beeping	Permanent (0.7 s on / 0.3 s off)
Alarm (not acknowledged)	Continuous beeping	Permanent

2.5 Intended Use

The controller may only be used for the applications described in the documentation and only in connection with devices, sensor cables or components from other manufacturers which have been approved or recommended by LIOS Technology. Correct, reliable operation of the product requires proper transport, storage, installation and commissioning as well as careful operation and maintenance.

OTS3-series controllers may be used for distributed temperature sensing in the applications described in section 2.6 except for fire detection. They are not evaluated by UL/ULC or VdS for use in fire detection.

LHD3-series controllers may be used as linear heat detectors for fire detection. Only LHD3-series controllers with DC power supply have been evaluated by UL/ULC and VdS for use in fire detection applications. The VdS approval comprises the standard LHD3-series controllers (laser operating at 1550nm) with and without EX-option as well as in the LHD3S-option controllers (laser operating at 1064nm) with and without EX-option. The UL approval only covers the standard LHD3-series controller without EX- and LHD3S-option.

Only OTS3/LHD3-EX models of controllers equipped with suitable sensor cables may be used for temperature sensing or heat detection in explosive areas. The OTS3/LHD3-EX controller itself must be operated outside the explosive area or inside an approved EX –d enclosure. The user has to verify that the classification given in the approval documents of the equipment comply with the requirements of his application. Restrictions given in the approvals need to be considered.

The controller is not suited for private use.



The controller may be used in dry, sheltered permanent installations. The controller is not suited for outdoor use without proper protection against weather.

The controller must be operated at supply voltages within the range indicated on the nameplate.

Non-authorized opening or modification of the controller, the sensor cable or the software as well as non-intended use lead to loss of any warranty and product liability.

2.6 Applications

This chapter presents typical applications of the controller. Further applications may be supported on demand (see section 6.2.1).

2.6.1 Fire Detection

LHD3-series controllers are approved linear heat detectors for fire detection in tunnels and buildings. They are also approved for fire detection of equipment operated at higher temperatures. Typical measuring times in fire detection last a few seconds only. A fire alarm is actuated within one to a few minutes depending on rate of temperature rise and in compliance with the respective national standards. Maximum temperature and local differential criteria also serve as triggers for fire alarm actuation. Besides alarm actuation, the controller also evaluates and indicates the size and propagation direction of a fire. Details are given in the national fire detection approvals of the controller.

The LHD3-series controllers have been evaluated by UL/ULC and the VdS for use in fire detection. Only alarm triggering by maximum temperature and rate-of-rise criteria has been evaluated by UL/ULC and VdS. The LHD3-series controllers have only been evaluated for use in ordinary locations by UL/ULC.

Typically, a temperature reading is acquired every 3 m for long objects such as road, cable and railway tunnels. For fire detection in buildings and other smaller objects, temperature readings are measured every 0.5 or 1 m.

After a fire, the sensor cable shall be inspected. In case of any damages, e.g. of the extruded sheeting, this part of the sensor cable shall be exchanged, even if the optical measurement shows no defect of the fiber.

To comply with UL/ULC standards, the sampling interval must be selected in accordance with the distance of cable used as outlined in the operation manual.

2.6.2 Power Cable Monitoring

The electrical current load of power cables is limited by the maximum allowed temperature of the boundary between the inner cable insulation and the conductor core. The present temperature of the conductor/insulation boundary does not only depend on cable design and current load, but also on cable laying, filling material, soil moisture, previous load, crossing with other cables and other heat sources. Cables may even be operated at a higher than the design load for a certain time because of their large thermal capacity.

Distributed temperature monitoring along a power cable using an OTS3-series controller enables safe cable operation at any location even at high loads. The optical fibre sensor cable may be incorporated into the shield of the power cable or attached externally. It is immune to electromagnetic interference.



Critical cable sections are easily located in the temperature profile. Conductor temperatures may be accurately calculated from sensor cable temperature and current load history using an optional real-time thermal rating (RTTR) software package. The RTTR software is also used for emergency calculations predicting critical conditions.

2.6.3 Downhole Monitoring

Thermal processes such as steam-assisted gravity drainage (SAGD) are increasingly used to produce highly viscous crude oil. Distributed temperature sensing using OTS3-series controllers considerably improves the efficiency and stability of such processes.

Downhole applications such as crude oil and gas production or geothermal energy usually show harsh environmental conditions at the sensor cable location. The sensor cable must withstand high mechanical forces, high temperatures, corrosive agents and hydrogen load. We offer special downhole cables (see Table 17) consisting of corrosion resistant, high-temperature materials with high strength. LIOS Technology optical fibres for such applications are composed of high-temperature polyimide coatings and a hydrogen-resistant pure silica glass optical core. These downhole fibres are long-term tested for hydrogen-resistance at elevated temperatures of up to 300°C.

Multichannel controllers for downhole applications may apply the special double-ended measuring mode to compensate for hydrogen-induced fibre aging.

2.6.4 Industrial Process Monitoring

Temperature monitoring is essential in a number of industrial processes. Typical application temperatures are quite high, and high spatial as well as temporal resolutions are wanted. OTS3-series controllers provide excellent spatial resolution for such applications and shortest measurement times. Our sensor cables for industrial applications are robust and suited for high temperatures (Table 17).

An example for the application of OTS3-series controllers in industry is the temperature monitoring in furnaces used for melting metals. The optical fibre sensor cable is installed in the thermal insulation of such furnaces in order to indicate wear or cracks of the insulation.

Further examples of OTS3-series controller's applications in industry are chemical plants, baking of composite materials, gas turbines, power plants and different steps of metal processing. Immunity against electromagnetic interference and excellent chemical stability of the optical fibre sensor cable are clear advantages of optical sensing compared to electrical measurements in a number of applications.

2.6.5 Leak Detection at Dams, Pipelines and Tanks

Most methods for leak detection used so far (video monitoring, mass flow measurements ...) are not capable of continuously monitoring lengthy objects with high spatial resolution. Distributed temperature sensing using an optical fibre sensor cable attached to the entire pipeline, dam or tank surface can accurately detect and locate leaks with short response time. Leak detection may be based on temperature changes caused by hot or cold liquids or gases such as heated oil in a pipeline or cold liquefied natural gas in a tank. Leakage of substances at ambient temperature may be detected by the Joule Thomson effect (cooling of gas during expansion) or by measuring heat conductivity of soil. The latter is done by using a hybrid sensor cable including optical fibres and electrical wires (Table



17). Electrical energy is used to heat the cable, and the temperature response of the cable is analysed in order to determine heat conductivity and thus moisture of the soil.

2.6.6 Environmental Monitoring

Distributed temperature sensing provides useful information on geophysical processes in rivers, streams, ground water, waste water channels, hot springs and other geothermal applications.

2.6.7 Radioactive Areas

We provide radiation hard optical fibres for distributed temperature sensing in radioactive areas. Those fibres may be used for years without degradation of sensing capabilities. The special JETEC sensor cable (Table 17) allows the exchange of the inner fibre optical cable by using a blow-in technology. Thus, long-term temperature monitoring in highly radioactive areas gets feasible without field service personnel entering those areas.

2.6.8 Explosive Areas

Optical fibres are in general excellently suited for applications in explosive areas. They do not carry electrical power and thus cannot generate electrical sparks. However, optical radiation emitting from broken fibres or sensor cables may heat materials and cause danger of explosion. Therefore, only LHD3-EX or OTS3-EX controllers may be used for temperature monitoring in explosive areas. Those controllers are inherently safe since the optical power within the sensor cable is securely limited to permitted levels. Controllers must be installed outside the explosive area unless they are enclosed in an approved EX -d containment. Details given in the approvals need to be considered. The OTS3/LHD3-EX-series controllers have not been evaluated by UL/ULC for use in hazardous locations.



3 Installation



Warning!

Caution!

Read this manual completely before installing the controller.

3.1 Hardware Installation

3.1.1 Unpacking the Controller

Check the controller package for visible damages directly after receipt. Check the shock label outside the package (Figure 13) before opening the package. A red shock indicator proves rough handling during transport. Note red shock indicator on the bill of lading. Report damages and red shock indicators immediately in writing and including pictures of the damage to the freight forwarder and to our Technical Support (see section 6.2.4).



Figure 11: Standard transport box



Figure 12: Optional heavy duty transport case



Figure 13: Shock label. A red tube in the centre would indicate rough handling

Unpack the controller at the commissioning location only. Check the controller and accessory for visible damages immediately after unpacking. Afterwards, check the completeness of the shipment using the packing list. Report damages and missing components in writing to our Technical Support (see section 6.2.4).

Ensure that no condensation (dewing) develops on or inside the controller after transport at low temperatures or after exposure to extreme temperature fluctuations. The controller must have acquired room temperature before it is put into operation. Do not expose the controller to direct radiation from a heater in order to warm it up. If dewing has developed, wait approximately four hours until the controller has dried completely before switching on the controller.

3.1.2 System Components and Tools

The following components are required for the installing the controller:

- Installation Manual
- Controller
- AC or DC power supply cable according to controller model
- Four M6x15 screws for rack mounting
- Charon3 software CD
- Optical fibre sensor cable(s)
- E2000/APC pigtail(s) for each optical fibre sensor cable
- Splice box(es) with or without grounding for each optical fibre sensor cable
- Termination box(es) with or without grounding for each optical fibre sensor cable
- Sensor cable termination kits if termination boxes are not available
- Sensor cable ties or clamps
- USB, Ethernet and/or serial (null modem) cable(s)
- Input and/or output cables if required

The following tools are required for installation of the controller:

- Torx T30 or Phillips 2 screw driver for mounting the controller
- Splicing equipment including fibre cleaver for installing sensor cable(s)
- Hot air gun if a termination kit will be used
- Fibre scope for ferrule endface inspection
- Fibre cleaning set including
 - lint-free optical wipes (e.g. Kimwipes EX-L)



- ultra-pure (99.9%) ethanol or iso-propanol solvent (We would recommend solvent for UV spectroscopy grade for residual-free cleaning.)
- ferrule mate cleaner if required (see section 3.1.6.4)
- Computer or Laptop compliant to requirements (see Operation Manual)

3.1.3 Controller Location

The controller must be installed in a dry and low-dust room, rack or outdoor box. Any condensation of moisture must be avoided.

The controller has only been evaluated by UL/ULC for ordinary indoor locations. The controller must not be thermally insulated during operation. For example, it must not be wrapped with foams, plastics films or other low thermal conductivity or air flow blocking materials. Free air circulation around the controller is essential if the operating temperature is higher than 50°C.

The controller may be installed inside a hermetic box, e.g. for explosion protection or in submarine applications, if sufficient heat transfer is guaranteed to maintain the required range of operating temperatures.

In case of an overheating and if the LON protocol is enabled (see Operation Manual), the controller sends an ENVIRONMENTAL CONDITIONS WARNING address point (> 60°C ambient temperature and / or too low heat transfer) and an ENVIRONMENTAL CONDITIONS ERROR address point (> 65°C). At 65°C, the controller also stops the measurement and activates the COLLECTIVE FAULT output. At even higher temperatures (> 70°C), the controller disconnects from the power supply. It starts up again if the temperature is within the permitted range.

The controller must not be installed in explosive areas. The specifications given in Table 21 apply to the operation of the controller.

Examples of applications where the use of the controller requires additional protection:

- Locations with a high degree of ionising radiation
- Locations with extreme operating conditions resulting from situations such as the following:
 - Corrosive vapours, gases, oils or chemicals
 - Electric or magnetic fields of high intensity
 - Greater shock and vibration levels
- Plants requiring special monitoring features, for example:
 - Elevator systems
 - Systems in especially hazardous rooms

3.1.4 Rack Mounting

3.1.4.1 Racks

The controller may be installed in a suitable 19-inch rack (Figure 14). The rack may also be used for installing the splice box, a computer, network switch, UPS and other devices. Forced air cooling or even air conditioning of the rack must be considered depending on the total power consumption of the devices.



Figure 14: 19-inch rack comprising controller, computer and UPS

3.1.4.2 Mounting position

The controller may be mounted in any vertical, horizontal or inclined position.

3.1.4.3 Fixation

Screw the controller tightly to the fixing brackets attached to the left and right of the front panel using the four M6 x 15 screws (Figure 15). The screws should be fastened crosswise.



Figure 15: Left and right screws locations

3.1.5 Connecting Electrical Cables

3.1.5.1 Power Supply



Warning!

Caution!

Check the supply voltage to be within the range given at the supply socket of the controller before connecting the power supply cable.



In fire detection applications according to pr EN 54-22, LHD3 controllers must be operated using an approved EN 54-4 DC power supply. Controller and power supply must be located in the same enclosure.

The Power Supply Cable DC 12-48V (Figure 16) provides a connector for the controller Sub-D3 WK3 socket at one end and three single leads at the other end. Connect the leads as follows:

- Brown lead: +12 to +48V DC¹
- Blue lead: 0V DC
- Green-yellow lead: Earth

The DC powered controller has an internal reverse-polarity protection. It does not operate with wrong polarity.

If you connect the controller to a 12 V battery supply, the battery voltage decreases while the battery is discharged. If the LON protocol is enabled (see Operation Manual), the controller sends an ENVIRONMENTAL CONDITIONS WARNING address point when a voltage of 11.4 V or lower is reached. At a voltage of 10.8 V or lower, the controller sends an ENVIRONMENTAL CONDITIONS ERROR address point and activates the COLLECTIVE FAULT output. The controller shuts down at 10 V and starts up again if at least the start-up voltage (Table 19) is applied. Please charge the battery when the ENVIRONMENTAL CONDITIONS WARNING address point is sent.



Figure 16: Power Supply Cable DC 12-48V

The optional AC power supply comprises a IEC320 C14 socket (Figure 10) to be connected to the standard Power Supply Cable 230V AC (Figure 17) or to an optional national Power Supply Cable AC.

¹ According to UL/ULC standards, the minimum operating voltage is + 14 VDC.



Figure 17: Power Supply Cable 230V AC, European Plug

3.1.5.2 Ethernet

An Ethernet cable (Figure 18) of category 5 or above may connect the controller to other devices using an Ethernet interface. Standard patch cables are used for connections to hubs or switches, whereas cross-over cables are used for direct connection to laptops and personal computers. Cross-over cables supplied by LIOS have a red sheath (Figure 19), whereas the LIOS patch cables have a light grey sheath (Figure 18). Cross-over cables from other suppliers may be identified by the different lead colour sequences inside the two connectors.



Figure 18: LIOS Ethernet patch cable



Figure 19: LIOS cross-over Ethernet cable

3.1.5.3 USB

Any standard USB cable (Figure 20) may connect the controller to a computer using the USB interface. The length of the cable should not exceed 3 m.

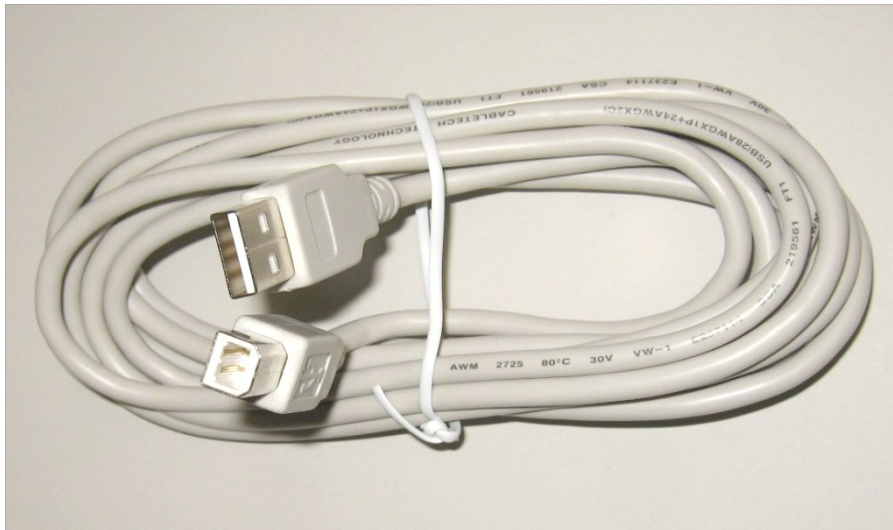


Figure 20: USB cable

3.1.5.4 Serial Interface

A shielded null modem cable (Figure 21) may connect the controller to another device using the EIA-232 (RS232) serial interface. The cable must have cross-over linked wires as indicated in Table 8. Serial cables without cross-over linking are not suitable. The length of the cable should not exceed 3 m.

Table 8: Null modem cable cross-over linking scheme

<i>Controller Signal</i>	<i>Controller Plug Pin</i>	<i>2nd Plug Pin</i>
DCD	1	4
RxD	2	3
TxD	3	2
DTR	4	1
Signal Ground	5	5
DSR	6	6



RTS	7	8
CTS	8	7
RI	9	9



Figure 21: Serial interface cable

3.1.5.5 Standard Outputs

The standard outputs are available at the OUTPUT socket on the rear panel (Figure 8). The pin assignments and output cable (Figure 22) lead colours are given in Table 9.

There are 10 programmable outputs available at the OUTPUT socket. The Charon3 software (see Operation Manual) is used to configure the type, assignment and normal status (open or closed) of each programmable output.

The COLLECTIVE FAULT output is closed during normal operation, and it is open in case of:

- power off,
- controller is not ready,
- no measurement is running,
- test mode is active
- fibre break or
- system fault.

The COLLECTIVE ALARM output is open if no alarm is active and closed in case of an active or not yet acknowledged alarm.

Outputs are galvanically isolated and not connected to a common ground.

Table 9: Pin and lead colour assignment of outputs

<i>Signal</i>	<i>Pin</i>	<i>Colour</i>	<i>Signal</i>	<i>Pin</i>	<i>Colour</i>
+ Output 1	1	white	- Output 1	14	brown
+ Output 2	2	green	- Output 2	15	yellow
+ Output 3	3	grey	- Output 3	16	pink
+ Output 4	4	blue	- Output 4	17	red



+ Output 5	5	black	- Output 5	18	violet
+ Output 6	6	grey/pink	- Output 6	19	red/blue
+ Output 7	7	white/green	- Output 7	20	brown/green
+ Output 8	8	white/yellow	- Output 8	21	yellow/brown
+ Output 9	9	white/grey	- Output 9	22	grey/brown
+ Output 10	10	white/pink	- Output 10	23	pink/brown
+ COLLECTIVE FAULT	11	white/blue	- COLLECTIVE FAULT	24	brown/blue
+ COLLECTIVE ALARM	12	white/red	- COLLECTIVE ALARM	25	brown/red
Not used	13				



Figure 22: Output cable

Figure 23 shows an example for connecting an alarm control system to outputs of the controller. The upper part of the figure shows the connection of a current monitored alarm input of the alarm control system to a normally open output of the controller. The alarm control system applies a voltage to its input and monitors the integrity of the cable by detecting the current flowing through the resistor R_p . In case of an alarm, the controller closes the output, and additional current flows through the resistor R_s . The alarm control system recognizes the alarm by the increase of current. For specifications of R_p and R_s check the manual of the alarm control system.

The lower part of Figure 23 shows the connection of an input of the alarm control system to a normally closed output of the controller. The alarm control system applies a voltage to its input and monitors the current flowing through the closed output of the controller. In case of a fault, the controller opens the output, and the alarm control system recognises the fault by detecting no current flow.

In principle, alarms may also be transmitted using normally closed outputs. However, interruption of the power supply of the controller or a break of the cable between both devices would trigger a false alarm at the alarm control system. Therefore, we strongly recommend only using normally open outputs for alarm transmission.

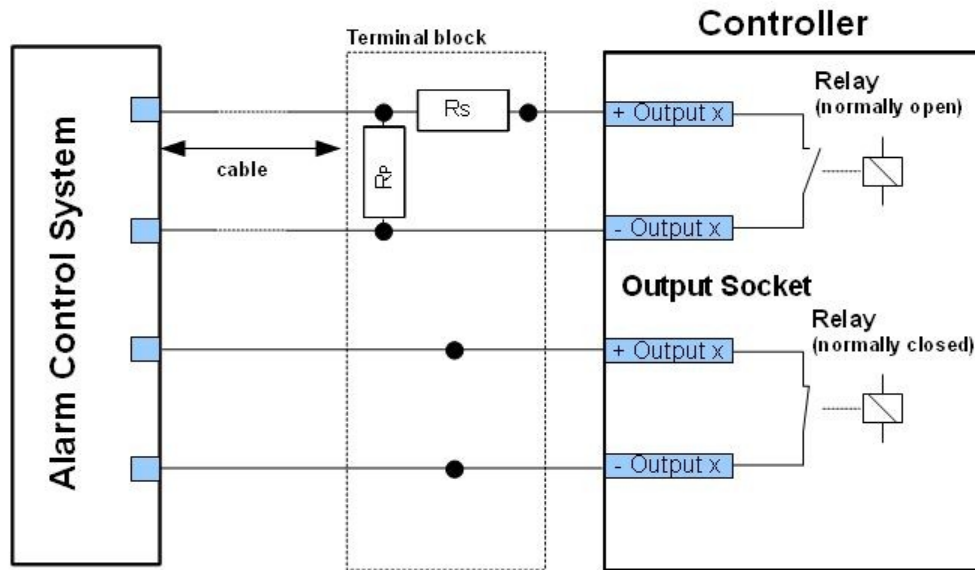


Figure 23: Example for connecting an alarm control system to output x (x = 1, 2, ...) of the controller

3.1.5.6 Extra Outputs

Extra outputs are available at the optional OUTPUT 1 to OUTPUT 8 sockets on the rear panel (Figure 9). Each socket connects to 12 outputs with pin assignment as given in Table 10. Extra outputs use the same type of output cable as the standard outputs. Table 11 gives the relative output number for each socket.

Table 10: Pin assignment of extra outputs

Signal	Pin	Signal	Pin
+ Output 1	1	- Output 1	14
+ Output 2	2	- Output 2	15
+ Output 3	3	- Output 3	16
+ Output 4	4	- Output 4	17
+ Output 5	5	- Output 5	18
+ Output 6	6	- Output 6	19
+ Output 7	7	- Output 7	20
+ Output 8	8	- Output 8	21
+ Output 9	9	- Output 9	22
+ Output 10	10	- Output 10	23
+ Output 11	11	- Output 11	24
+ Output 12	12	- Output 12	25
Not used	13		

The output numbers used by the Charon3 software (see Operation Manual) for each socket are summarized in Table 11. The Charon3 output number is calculated as follows:

$$\text{Output number} = 10 + (\text{number of OUTPUT socket} - 1) * 12 + \text{relative output number},$$

e.g. output 3 on OUTPUT 4 socket has the output number $10 + 3 * 12 + 3 = 49$.

Table 11: Charon3 output number range for each output socket

Output Socket	Output Numbers
---------------	----------------



OUTPUT	1 to 10
OUTPUT1	11 to 22
OUTPUT2	23 to 34
OUTPUT3	35 to 46
OUTPUT4	47 to 58
OUTPUT5	59 to 70
OUTPUT6	71 to 82
OUTPUT7	83 to 94
OUTPUT8	95 to 106

3.1.5.7 Standard Inputs and 24 V Output

The four standard inputs are available at the INPUT socket on the rear panel (Figure 8). Pin assignment and input cable (Figure 24) lead colours are given in Table 12. Standard inputs refer to a common ground. They are activated by applying the specified voltage (Table 22) between the input pin and ground. Polarity must be considered. Inputs will not operate with wrong polarity.

A 24 V DC auxiliary voltage for switching the standard inputs is provided at the same Sub-D9 socket. The voltage refers to ground. The maximum current is 20 mA. The voltage output is protected by a self-resetting fuse.

Table 12: Pin and lead colour assignment of standard inputs

Signal	Pin	Colour	Signal	Pin	Colour
Input 1	1	White	Ground	6	Brown, grey
Input 2	2	Green	Ground	7	Blue
Input 3	3	Yellow	Ground	8	Black
Input 4	4	Pink	+ 24 V DC output	9	Red
Not used	5	Violet			



Figure 24: Input Cable (4 Inputs, 24V DC Output)

Figure 24 shows examples for connecting relay outputs of fire detectors to inputs of the controller. Since both, the output and the input are relays, a voltage source is required. For the standard inputs, you can use the +24 V DC from the controller supplied at the standard inputs socket as shown in the upper part of the figure. You may also use a 24V DC voltage supply from the fire detector or another



device as shown in the middle part of the figure. Extra inputs can only be used with external voltage supply from the fire detector or another device as shown in the lower part of the figure.

Caution!

The 24V voltage supply of the controller can only supply current for up to 4 inputs. Do not connect higher numbers of inputs to this voltage supply!

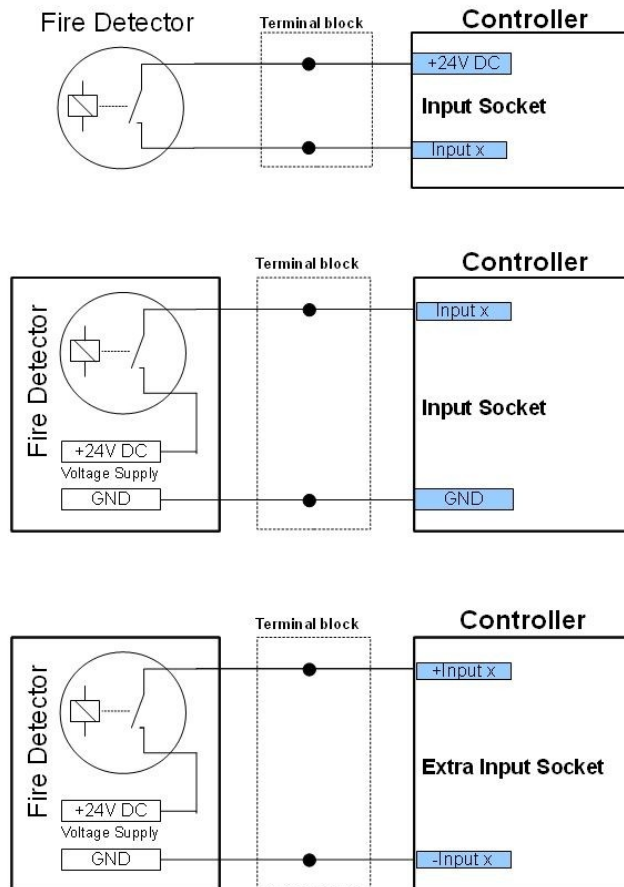


Figure 25: Examples for connecting fire detectors to input x (x = 1, 2, ...) of the controller

3.1.5.8 Extra Inputs

Extra inputs are available at the optional INPUT 1 to INPUT 3 sockets on the rear panel (Figure 9). Each socket connects to 12 inputs with pin assignment and input cable (Figure 26) lead colours as given in Table 14. The table gives the relative input number for each socket.

The input numbers used by the Charon3 software (see Operation Manual) for each socket are summarized in Table 13. The Charon3 input number is calculated as follows:

Table 13: Charon3 input number range for each input socket

Input socket	Input numbers
INPUT	1 to 4
INPUT1	5 to 16
INPUT2	17 to 28



Input number = 4 + (number of INPUT socket -1)*12 + relative input number,

e.g. input 3 on INPUT 2 socket has the input number 4 + 12 + 3 = 19.

Extra inputs are galvanically isolated and not connected to a common ground. They are activated by applying the specified voltage (Table 22) between both respective input pins. Polarity must be considered. Inputs will not operate with wrong polarity.

Table 14: Pin and lead colour assignment of extra inputs

Signal	Pin	Colour	Signal	Pin	Colour
+ Input 1	1	white	- Input 1	14	brown
+ Input 2	2	green	- Input 2	15	yellow
+ Input 3	3	grey	- Input 3	16	pink
+ Input 4	4	blue	- Input 4	17	red
+ Input 5	5	black	- Input 5	18	violet
+ Input 6	6	grey/pink	- Input 6	19	red/blue
+ Input 7	7	white/green	- Input 7	20	brown/green
+ Input 8	8	white/yellow	- Input 8	21	yellow/brown
+ Input 9	9	white/grey	- Input 9	22	grey/brown
+ Input 10	10	white/pink	- Input 10	23	pink/brown
+ Input 11	11	white/blue	- Input 11	24	brown/blue
+ Input 12	12	white/red	- Input 12	25	brown/red
Not used	13				



Figure 26: Input Cable (12 Inputs)

3.1.5.9 External Sensors

External sensors may be connected to the controller using the optional SENSORS socket on the rear panel and an external sensors cable (Figure 27) or using Ethernet. The SENSORS socket pin and external sensors cable lead colour assignments are given in Table 15.

Table 15: Pin and lead colour assignment of external sensors

Signal	Pin	Colour	Signal	Pin	Colour
+R1	1	white	-R1	11	brown



+RS1	2	green	-RS1	12	yellow
+R2	3	grey	-R2	13	pink
+RS2	4	blue	-RS2	14	red
+5VDC	5	black	Ground	15	violet
Earth	A1	grey/pink	Earth	A1	red/blue
+12VDC	6	white/green	+24VDC	16	white/yellow
+I1	7	white/grey	-I1	17	grey/brown
+I2	8	white/pink	-I2	18	pink/brown
+U1	9	white/blue	Ground	19	brown/green, yellow/brown
+U2	10	white/red	Ground	20	brown/blue, brown/red



Figure 27: External sensors cable

Auxiliary voltages of 5V, 12V and 24 V DC for sensor operation are provided at the SENSORS socket. The voltages refer to ground.

Each of the two pin groups +/-R1, +/-RS1, respectively +/-R2, +/-RS2 may be connected to a Pt100 temperature sensor in 2-, 3- or 4-wire configurations. +/-R1 and +/-R2 pins are the sensor current supply line pairs, and the +/-RS1 and +/-RS2 pins are the sensor resistance sensing line pairs. Example connection schemes are given in Figure 28.

Do not connect sensors to external power supplies.

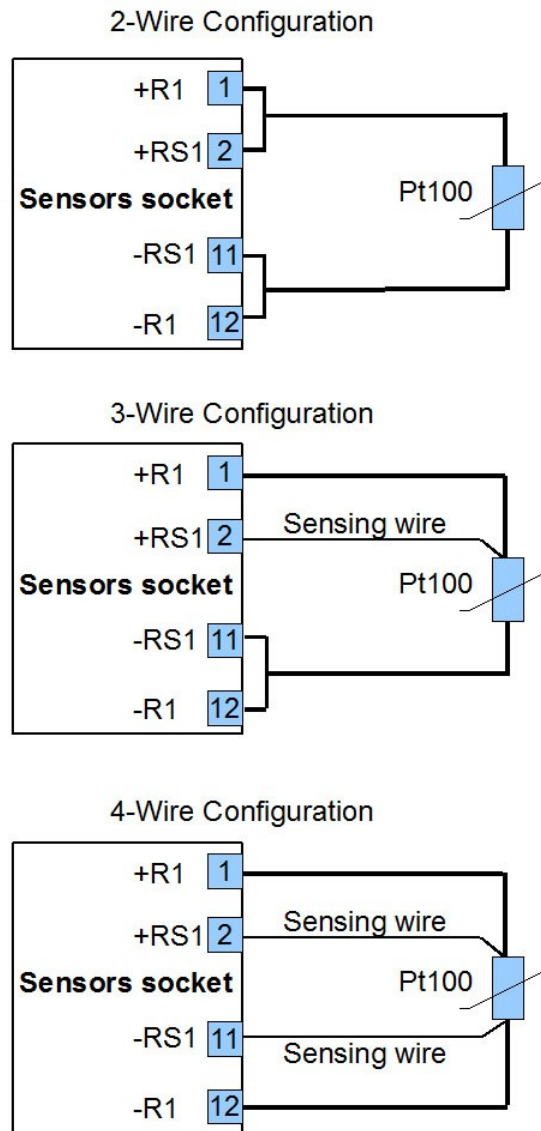


Figure 28: Connection schemes for a Pt100 temperature sensor in 2-, 3- and 4-wire configurations

Each of the two pin groups +/I1 and +/I2 may be connected to a 0 to 20 mA (Figure 29) or a 4 to 20 mA current sensor. The 4 to 20 mA sensor requires an auxiliary voltage provided by the controller using the connection scheme as shown in Figure 30.

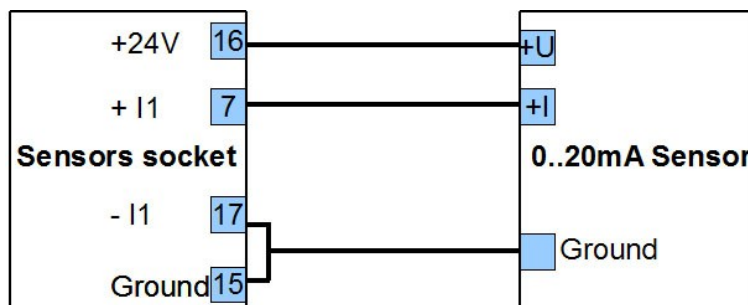


Figure 29: Connection scheme for a 0 to 20 mA current sensor using a supply voltage of 24V

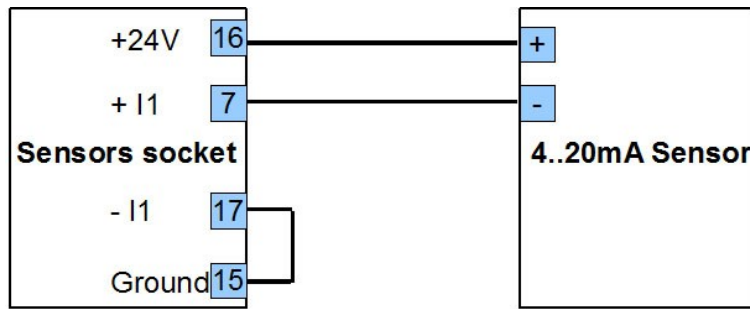


Figure 30: Connection scheme for a 4 to 20 mA current sensor using a supply voltage of 24 V

Up to two 0 to 2, 5 or 10 V voltage sensors can be connected to +U1 and +U2 and ground (Figure 31).

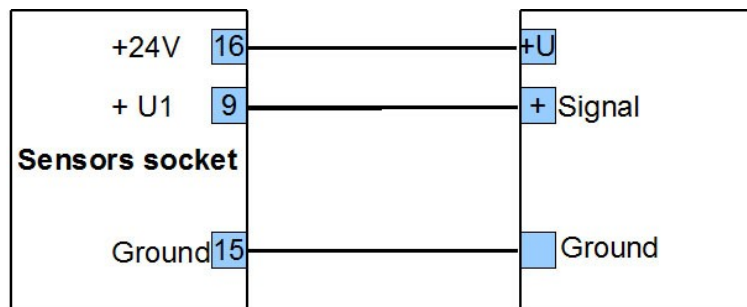


Figure 31: Connection scheme for a 0 to 10 V voltage sensor using a supply voltage of 24 V

Digital fieldbus sensors may be connected to the controller over Ethernet using the optional fieldbus protocol driver.

3.1.6 Connecting Sensor Cable(s)

3.1.6.1 Fibre Type

Each OTS3-series controller is suited for a certain fibre core diameter indicated by two digits in the controller model (see Figure 2 and section 2.1.4):

Table 16: Fibre types

Controller model -yy	Fibre type	Fibre core diameter	Fibre N.A.	Index profile
-09	Single-mode	9 µm MFD @ 1310 nm	0.14	step
-50	Multimode	50 µm	0.20	graded
-62	Multimode	62.5 µm	0.27	graded
None (LHD3-series)	Multimode	62.5 µm	0.27	graded

The LHD3-series controllers are suited for graded index fibres with a core diameter of 62.5µm and a N.A. of 0.27.

Only sensor cables, fibres and pigtails with matching core diameter and N.A. must be connected to the controller.

3.1.6.2 Connector Type

Caution!



The controller comprises up to 16 optical ports for E2000/APC (angled physical contact) connectors with 8° angular polished ferrule endfaces. Do not use other types of connectors.

The best measuring performance is achieved by using pre-assembled E2000/APC connectors (so-called pigtailed) that are spliced to the sensor cable. The splices need mechanical protection by a splice cassette and a splice box.

We would not recommend field assembly of connectors because of the lower precision of ferrule endfaces and thus higher attenuation leading to reduced measuring performance of the controller.

Do not use patch cables for connecting other connector types or straight PC (physical contact) connectors, because of the additional attenuation and return loss.

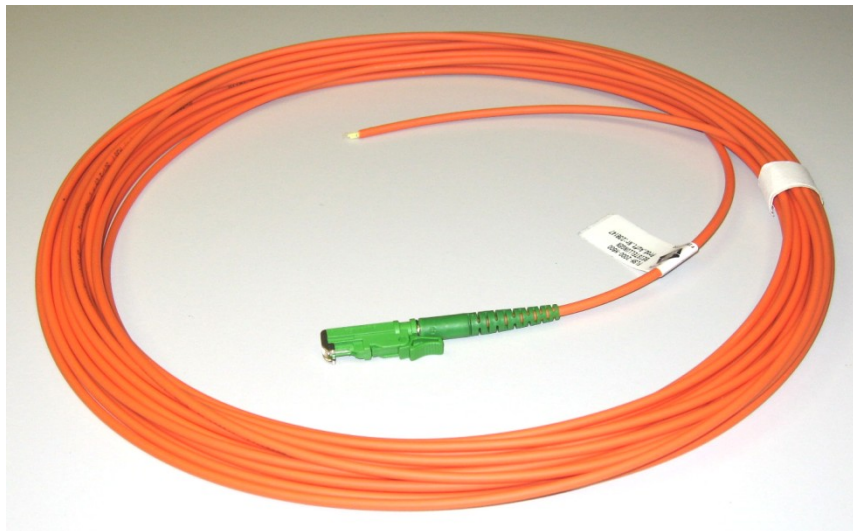


Figure 32: Pigtail with E2000/APC connector

3.1.6.3 Laying Pigtails and Sensor Cable(s) in the Rack

The pigtail(s) and splice box(es) must be laid and mounted in a way that securely avoids any strain on pigtails, splices and E2000/APC connectors as well as any kinking of the pigtail or the sensor cable (Figure 33). Bend excessive lengths of pigtail and sensor cable to loops with sufficient diameter (> 100 mm) and loosely attach them to the rack or other solid structures connected to the controller. Loosely fastened plastics cable ties may be used for attaching the loops. Avoid any crushing of the sensor cable or pigtail.

Mount the splice box in close proximity to the controller. For metal containing sensor cables, use a splice box with grounding (see section 3.2.7). Connect the grounding bolt of such splice boxes to an earth cable being connected to an appropriate earthing point of the facility.



Figure 33: Mounting of splice boxes inside a rack

3.1.6.4 Cleaning and Inspection of Connectors



Warning!

Caution!

Switch off the controller before any direct viewing with or without optical instruments on fibre ends, splices or optical fibre connectors. In addition, the warning label shown in Figure 1 has to be placed close to the mains switch of the controller. Use only CE marked fibre scopes equipped with a 1550 nm DIR LB3 laser safety filter according to EN 207.

Check the ferrule endfaces of any connector for dirt and damages using a fibre scope (Figure 34).



Figure 34: Fibre scope with 200x magnification, built-in laser filter and LED illumination

If any dirt is present on the endface of a ferrule (Figure 35), carefully open the protective flap of the E2000 connector. Moisten a lint-free optical wipe slightly with ultra pure solvent, and carefully clean the endsurface of the ferrule (Figure 36). Repeat the procedure using a dry wipe to remove any residues of the solvent. Check the endface of the ferrule again using the scope. If any dirt remains, the cleaning procedure must be repeated.

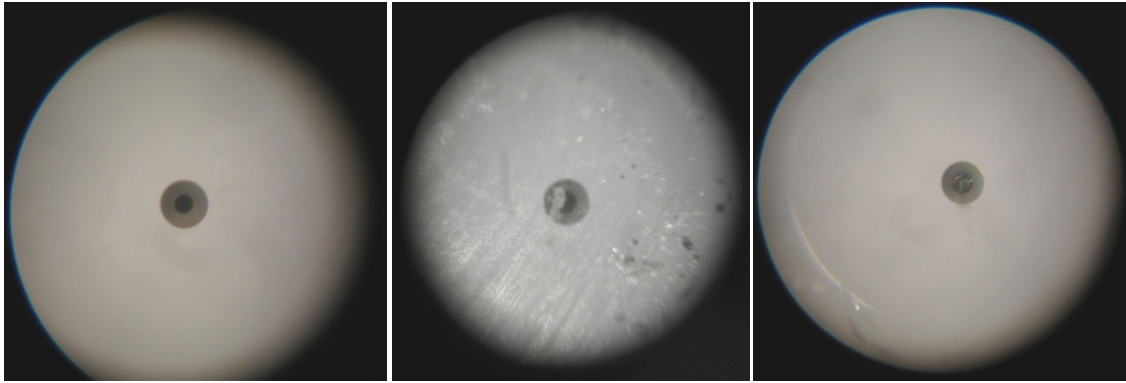


Figure 35: Views of clean (left), contaminated (middle) and damaged (right) pigtail endfaces

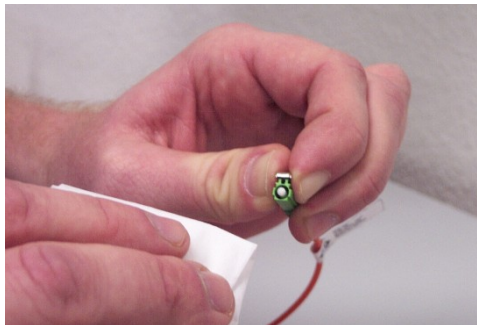


Figure 36: Cleaning of a ferrule endface of an E2000/APC connector using an optical wipe

Pigtails must be replaced if there is any damage or non-removable dirt on the endface of the fibre core (Figure 35). A few slight scratches on the fibre cladding or endface of the ferrule are acceptable if the attenuation and return loss of the connector are within specification (see sections 3.2.2 and 6.1). The pigtail must also be replaced if there is any sever damage (digs, excavations, deep scratches ...) or non-removable dirt on fibre cladding or endface of the ferrule (Figure 35).

We ship controllers with clean internal pigtail(s) being terminated by dummy connector(s) (Figure 37) plugged into the port(s). Dummy connectors must only be removed shortly before plugging a pigtail into the port. No cleaning of the internal pigtail(s) is required for new controllers whose ports have been permanently terminated by dummy connectors.



Figure 37: Dummy connector



A cleaning of the endfaces of internal pigtails is required before plugging any pigtail into a previously used or non-permanently terminated port. Use a ferrule mate cleaner (Figure 38) or equivalent tool for cleaning the internal pigtails. Follow instructions of the cleaner. Stick type cleaners are less reliable and may only be used if a fibre scope suited for internal pigtails is available for subsequent inspection. If any dirt remains, the cleaning procedure must be repeated.



Figure 38: Ferrule mate cleaner

3.1.6.5 Plugging-In Optical Connectors



Warning!

Caution!

The controller power must be switched off or the measurement must be stopped. Check that the “LASER ON” LED is off.

After proper cleaning, carefully insert the E2000/APC connector into an optical port on the rear side of the controller until it securely locks in place. Unused E2000/APC ports must stay terminated using dummy connectors.

3.2 Sensor Cable Installation

3.2.1 Cable Selection

Several sensor cable designs are available for different applications. Examples are given in Table 17.

Table 17: Examples for sensor cables

<i>Sensor Cable model</i>	<i>Diameter</i>	<i>Design / Properties</i>	<i>Application</i>
<i>Plastics cables</i>			
JETEC Cable	2.0 mm	Elastic plastics cable for blow in laying in 10 mm micro ducts	Nuclear power plants
Metal Free FRNC Cable	4.0 mm	Flame retardant plastics cable	Indoor, tunnels, fire detection (evaluated by UL/ULC and VdS)
Metal Free PA Cable	4.0 mm	Robust plastics cable	General
Metal Free Harsh Environment Cable	4.4 mm	Chemically inert plastics cable with high temperature fibres	Transformers, chemical plants
Underground Cable	7.0 mm	Robust plastics cable	Outdoor, cable ducts, buried
ADSS Cable	16.3 mm	All dielectric self supporting aerial cable	Overhead transmission lines
<i>Metal containing cables</i>			
FIMT 1.8	1.8 mm	Fibres in metal tube	Power cable monitoring
Double FIMT 2.1	2.1 mm	Robust and chemically inert design (fibres in double metal tube)	Chemical plants, industrial ovens, liquid natural gas
Stranded wire 2.1	2.1 mm	FIMT with stranded stainless steel	Chemical plants, industrial



		wires	ovens, liquid natural gas
Harsh Environment Cable	2.4 mm	High temperature fibres in corrosion resistant metal tube with chemically inert plastics sheath	Chemical plants, metal processing
Downhole ¼ Cable	3.2 mm	High temperature fibre in double metal tube	Oil industry, geothermal, liquid natural gas
Fast Response FRNC Cable	4.0 mm	Metal cable with flame retardant plastics sheath, rodent resistant	Indoor, tunnels, fire detection (evaluated by UL/ULC and VdS)
Silicone Cable	5.0 mm	High temperature fibre in metal tube with silicone sheath	Industrial ovens, chemical plants
Downhole ¼ Cable	6.4 mm	High temperature fibre in double metal tube with chemically inert plastics sheath	Oil industry, geothermal
Underwater Cable	7.9 mm	Robust, high density, low wear cable for repeated laying	Environmental monitoring
EX Protection Cable	4.0 mm	Metal cable with conductive plastics sheath	Explosive areas with dust
Hybrid Cable	12.9 mm	Hybrid cable for measuring response to electrical heating (heat conduction)	Dams, dykes

A full list of sensor cable models is available at LIOS Technology. We also design custom cables on request.



Figure 39: Sensor cable Metal Free FRNC



Figure 40: Sensor cable Fast Response FRNC

3.2.2 Fibre and Sensor Cable Recommendations

For OTS3-series controllers, the fibre and sensor cable specifications given in Table 18 are essential to obtain the controller performance as indicated in the datasheets. In fire detection, LHD3-series controllers must use approved LIOS Technology sensor cables having less than 0.75 dB/km attenuation @ 1300 nm after installation.



Table 18: OTS3 fibre and sensor cable specifications

Fibre	Attenuation	Bandwidth @ 1300 nm	Standard
SM 9/125	< 0.22 (+ 0.03) ² dB/km @ 1550 nm		ITU-T G.652 Table C or D
MM GI50/125	< 0.50 (+0.05) dB/km @ 1300 nm < 0.28 (+0.07) dB/km @ 1550 nm	> 1000 MHz*km	
MM GI62.5/125	< 0.70 (+0.05) dB/km @ 1300 nm < 0.33 (+0.07) dB/km @ 1550 nm	> 500 MHz*km	

A variation of attenuation along the installed sensor cable may make calibration difficult. Avoid such variations by use of similar or identical fibre qualities throughout the whole sensor cable, strain-free sensor cable laying and proper fibre splicing.

Weaker specified fibres may be used for OTS3-series controllers if unavoidable. A higher attenuation affects the distance range and temperature resolution of the controller. A lower bandwidth of fibres affects the spatial resolution at larger distances.

The fibre and cable sections must be connected using fibre splices. Do not use detachable connectors. Proper splices between fibres have less than 0.05 dB attenuation. If a detachable connector cannot be avoided, use an APC connector type with an attenuation below 0.3 dB and a return loss of 40 dB or higher (multimode) or with an attenuation below 0.1 dB and an return loss of 85 dB or higher (single-mode). An attenuation caused by connectors and splices may affect distance range and temperature resolution of the controller.

3.2.3 Handling of Sensor Cables

Caution!

Optical fibres are made of glass and are thus brittle. For sensing applications, the fibres are protected by various types of sensor cables making the sensing element much more robust than a bare optical fibre. Nevertheless, certain restrictions must be considered in handling optical fibre sensor cables. Each sensor cable is specified for minimum bending radius, maximum tension force and maximum crush impact. Choose a proper sensor cable being robust enough for your application. The sensor cable specifications must be considered in laying. Avoid too tight bending, any kinking, too high pulling forces, strong crushing and too tight fixing of the cable! Pull less robust sensor cables by hand only.

Pigtails provide a weaker mechanical protection than a sensor cable. Thus, the pigtails must be handled with special care. Pull and fix the pigtails very gently only. Never pull out a latching connector by pulling the fibre pigtail only.

Improper handling of fibres, pigtails and sensor cables may lead to excessive attenuation and even immediate or delayed fibre breaks. The fibre breaks and damages are not necessarily visible from the outside and may only be detected during use of the sensor cable.

² Attenuation values in parenthesis indicate the permitted additional attenuation due to cable manufacturing and installation.



The sensor cables are delivered on cable drums. The drum sheathing must only be removed shortly before cable laying. The cable drums must permanently stand upright to avoid damage during handling and storage. Empty cable drums should not be sent back to LIOS Technology since they are one-way packing materials.

3.2.4 Laying of Sensor Cables

3.2.4.1 Cable Length

The total active sensor cable length must be shorter than the distance range of the controller. When calculating the total sensor cable length, cable lengths outside the measurement section must be taken into account. These are:

- Feed cable from controller to measurement section,
- Connection cables and fibres,
- Avoidance of obstacles,
- Test and reference lengths,
- Height differences and
- About 5 m of fibre length in the controller cabinet.

In addition to the active sensor cable length, a few sampling intervals (e.g. 20 m at a sampling interval of 3 m) of inactive fibre length are required at the end of the measurement section to suppress influences of a possible fibre end reflex. The controller accepts an inactive fibre length of up to 100 m in addition the distance range. By this, proper temperature readings are obtained for the full distance range. An optional license is required for the use of larger inactive lengths.

If possible, use the controller with a minimum total fibre length of at least 100 m.

3.2.5 Laying Plan

The laying plan of the sensor cable should be outlined in a table. Meter markings of the cables at the object beginning, splicing locations, zone boundaries, etc., should be registered in the laying plan during laying.

Consider test sections for regular testing of controller function using cable heating equipment. Place test segments outside the monitored cable section to enable testing without deactivation of alarm transmission.

Consider a test heating for the determination of the location of zone boundaries. The length of the optical fiber may slightly vary from the meter marking on the sheathing of the sensor cable.

3.2.6 Laying Plans for Fire Detection

Applications according to EN 54-22, the standard DIN VDE 0833-2, or the guideline VdS 2095 on the installation of automatic fire detection systems in buildings require a detailed analysis of the instructions given in the standards and the guideline.

Basically, the sensor cable must be installed at the highest point within a room or a tunnel with a distance of 5 to 20 cm from the ceiling.

According to the standard DIN VDE 0833-2, the maximum height for mounting the sensor cable of a linear heat detector is 9 m for response class A1 and 7.5 m for response class A2.



According to the guideline VdS 2095, the maximum height for mounting the sensor cable of a linear heat detector is 7.5 m for response class A1N and 6 m for the other response classes. For meander-like laying schemes in rooms larger than 30 m² and with a roof slope of less than 20°, the maximum horizontal distance of any point of the ceiling to the sensor cable must be less than 3.5 m. The horizontal distance between parallel sensor cables must be less than 7 m and the distance between sensor cables and walls must be between 0.5 and 3.5 m. Larger distances specified in the guideline may be applied in smaller rooms and for steeper roofs.

When installed in the United States, the LHD3-series controllers are to be installed in accordance with NFPA No. 70 “National Electric Code” and NFPA No. 72 “National Fire Alarm and Signalling Code”.

When installed in Canada, the LHD3-series controllers are to be installed in accordance with:

- ULC-S524 Standard for the Installation of Fire Alarm Systems, and CSA 22.2
- The National Building Code of Canada
- The Canadian Electrical Code, Part 1
- The National Fire Code of Canada

For the fire detection in tunnels, a maximum horizontal distance of 6 m between any monitored location and the sensor cable is considered to be acceptable, if the sensor cable is installed at the maximum height mentioned above. Thus, a tunnel with two road lanes and a width of up to 12 m may be monitored using a single sensor cable. Additional sensor cables are required for wider tunnels and according to national fire safety regulations.

Redundant fire detection systems are required by most fire detection standards. The easiest way to achieve full redundancy is to measure the two fibres of a sensor cable from the opposite ends of the cable using two controllers. A redundancy with respect to the sensor cable only is achieved by measuring the two fibres of a cable loop from the opposite ends using a single controller.

If an unambiguous mapping of events within fire compartments is required, we recommend using an extra length of sensor cable in between the alarm zones of the fire compartments (see Operation Manual). Accordingly, a respective extra length of sensor cable has to be laid at the boundary of two fire compartments. The extra length should be three sampling intervals on each side of the boundary, i.e. in total six sampling intervals, but not less than 2 m. Check the sampling interval of the measuring parameter set to be used and calculate the extra length accordingly. Sampling intervals are between 3 m and 0.25 m depending on controller model, sensor cable length and application. Thus, extra lengths between 18 m and 2 m are required. Depending on the application, the extra length may not be covered by a zone or covered by an inactive zone, a zone without alarm parameter set or an alarm zone not assigned to a fire compartment.

The laying plan has to consider possible sources of false alarms. Keep a safe distance from heat sources such as lamps and avoid locations with sudden temperature changes. If compliance with EN 54-22 is not required, less sensitive alarm parameter sets (see Operation Manual) may be used to prevent false alarming in such locations.

3.2.7 Connecting Sensor Cables

Connect sensor fibres in cable sections, pigtails and terminations by fusion splicing. A suitable splicing apparatus and preparation tools including fibre cleaver are essential. The splicing of fibres requires a



certain level of experience. Please ask a local telecommunications service company for support if you do not have sufficient experience.

Lay splices in a splice cassette (Figure 41) and splice box with (Figure 42) or without grounding (Figure 43). We strongly recommend using the robust splice boxes in any rough environment such as tunnels, cable ducts and outdoor installations. Even in buildings, the protection of the sensitive splice in a box is advantageous.



The splice boxes with grounding are required for metal containing sensor cables (Table 17) if there is any risk of interaction with electricity. Please consider not only electrical cables and wires, but also lightning, ESD and electromagnetic induction.



Figure 41: Splice cassette



Figure 42: Splice box without grounding and splice cassette

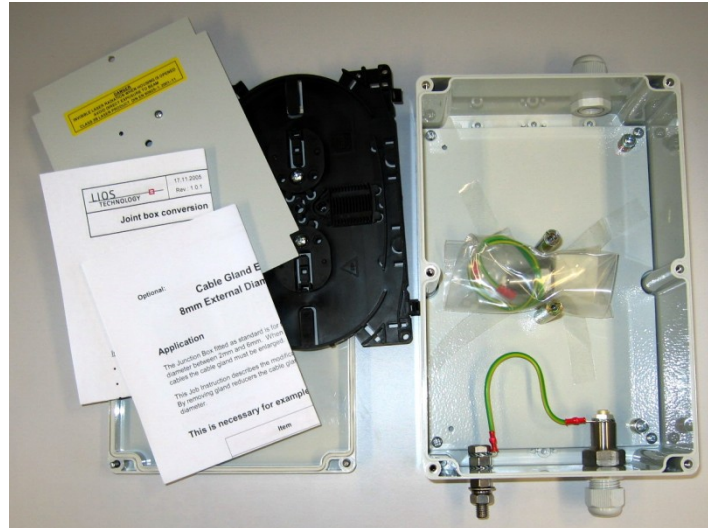


Figure 43: Splice box with grounding and splice cassette

Follow the procedure below for laying a splice in a splice box:



Warning!

- Switch off any controller connected to the sensor cable. Place a warning label close to the controller (see section 1.4).
- Fix the splice box at a solid structure.
- Clean sensor cable ends of rough dirt.
- Feed sensor cable and pigtail into the splice box through the cable feeds. Pull at least 2 m of cable and pigtail through the feeds. The cable and pigtail lengths outside the box must be sufficient to safely avoid any strain, tight bending with less than 60 mm radius and kinking.
- Remove 2m of sensor cable sheath (outer plastics layer) and armour.
- Remove the stainless steel or plastics tube on the same end, but with 5cm excess length measured from the end of the sheath. Take care not to damage the fibres.
- Earthen the stainless steel tube, if present, using the multi-contact cylinder of the feed through.
- Remove 2m of pigtail sheath. Take care not to damage the fibres.
- Feed unsheathed fibres from cable and pigtail into the splice cassette.
- Let sensor cable and pigtail sheath extend for about 5 cm into the splice box and relieve strain at the support points provided with cable ties.
- Splice fibre ends.
- Stabilise the splice using a hot-melt adhesive or crimp protection sleeve.
- Fasten the protection sleeve in the sleeve holder of the cassette, and lay fibres in the splice cassette. Consider a minimum bending radius of 60 mm.
- Close the splice cassette carefully and fix the cassette cover.
- Close the cover of the splice box and fix it.

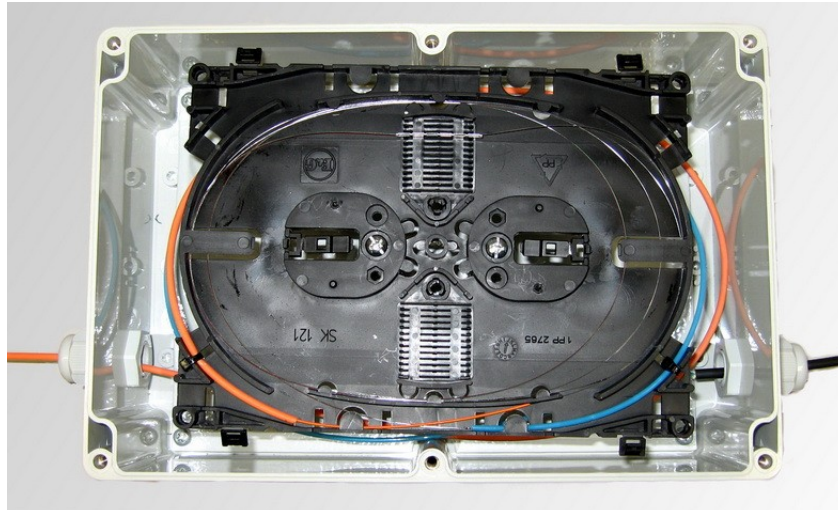


Figure 44: Splice box with cable and pigtail

3.2.8 Termination of Sensor Cables

A length corresponding to a few sampling intervals (20 m at a sampling interval of 3 m) of the sensor fibre at the remote end should not be used for temperature monitoring or heat detection because of a possible optical interference from the fibre-end. Correspondingly, the last 20 meters of sensor cable should be deposited in a cable duct, or the same length of bare sensor fibre should be deposited in a proper splice (Figure 42, Figure 43) or termination box (Figure 45). The fibre-end must be properly terminated to avoid excessive back reflection of laser radiation into the sensing fibre. Otherwise, measurement accuracy close to the fibre-end might be affected.

We supply optional termination boxes (Figure 45) including properly terminated fibres with a length of 20 meters each. The fibre lengths inside the box or cassette just have to be spliced to the sensor fibres for correct termination.

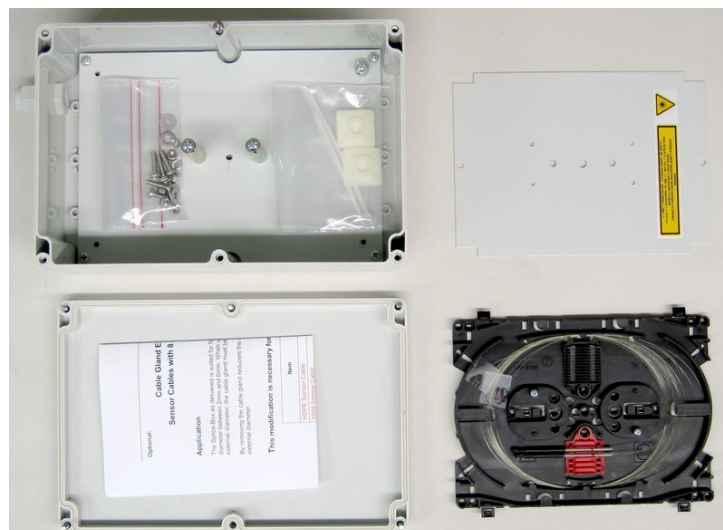


Figure 45: Termination box

The termination kit (Figure 46) may be used if no termination box is available. The kit is used as follows:

- Remove 50 mm of the cable sheath.



- Remove 45 mm of the inner cable tube.
- Remove 20 mm of the fibre coating.
- Clean the fibre.
- Cut the fibre end almost diagonally using the fibre cleaver. The resulting cleaving angle must be larger than 20°.
- Slide the fibre end into a 60 mm long hot-melt splice protection sleeve. Locate the fibre end in the middle of the splice protection sleeve. Melt the adhesive using a hot air gun (220°C).
- Slide the black heat-shrinkable tubing 25 mm over the cable sheath. The heat-shrinkable tubing should protrude 25 mm over the splice protection sleeve. Fix the tubing using the hot air gun. Press the end of the hot tubing slightly together to securely seal the end.



Figure 46: Termination kit comprising hot-melt splice-protection sleeves and black heat-shrinkable tubings

3.2.9 Repair of Sensor Cables



Warning!

In case of any damage to a sensor cable, switch off the controllers connected to the cable as soon as possible.

Cut out the damaged sensor cable section and replace it with a new cable length. Set two splice boxes as described in section 3.2.7.

Lay the replacement sensor cable with 3 to 5 m extra length on each end as compared to the damaged cable. Set the two splice boxes in the middle of the overlaps to ensure that sufficient fibre length is available for splicing (Figure 47).

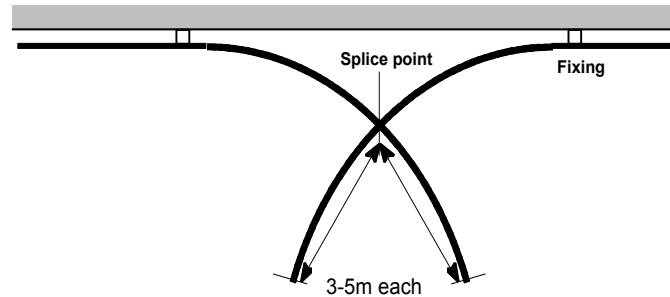


Figure 47: Cable laying for repair

3.2.10 Ceiling Mounting

Ceiling mounting is the typical way of sensor cable installation in fire detection applications. For fire detection, use only sensor cables approved for fire detection with the controller.

The sensor cable must be placed in about 5 cm distance from the ceiling to minimize heat conduction. It also needs a line of sight to the monitored object to be sensitive for heat radiation. Use a single, continuous length of sensor cable if anyhow possible since splicing reduces measurement quality and is an extra effort. Keep a safe distance from heat sources such as lamps or heatings to avoid false alarms due to heating.



Warning!

Keep a safe distance from high voltage lines as they are present in railway tunnels.

The sensor cable may be fixed using different kinds of fastenings such as cable ties (Figure 48), nail plug clips (Figure 49), cable clips (Figure 50) or tube clamps (Figure 51). A tension rope may be used for secure mounting of a sensor cable without strain on the cable (Figure 52). Only use UV stabilized cable ties for outdoor applications.

Always avoid excessive crushing of the cable by strong tightening of the fastenings. As a rule of thumb, a fixed cable should be slightly pullable through the fastenings with a normal level of force. Consider using soft rubber sheets to reduce crushing of the cable by cable ties. We would recommend a distance of 0.75 to 1 m between the fastenings for mounting in a straight line (Figure 53). A tension rope may be used to support a sensor cable with limited number of wall or ceiling fastenings (Figure 52).



Figure 48: Sensor cable mounted with cable ties

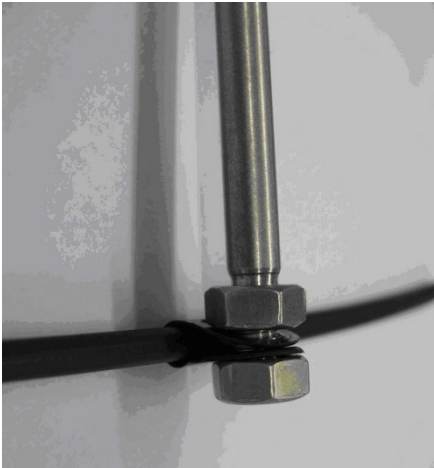


Figure 49: Nail plug pin

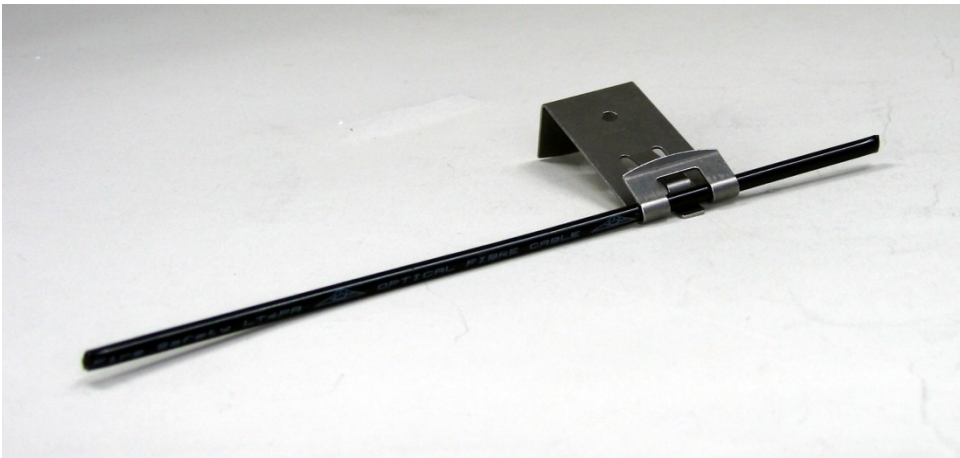


Figure 50: Cable clip

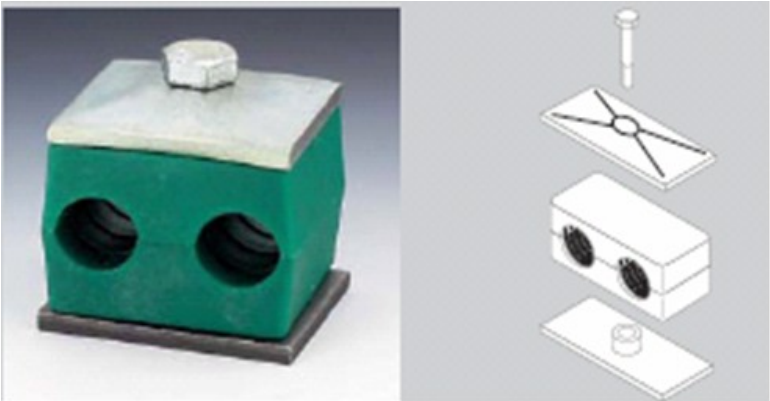


Figure 51: Tube clamp for fixing larger diameter sensor cables

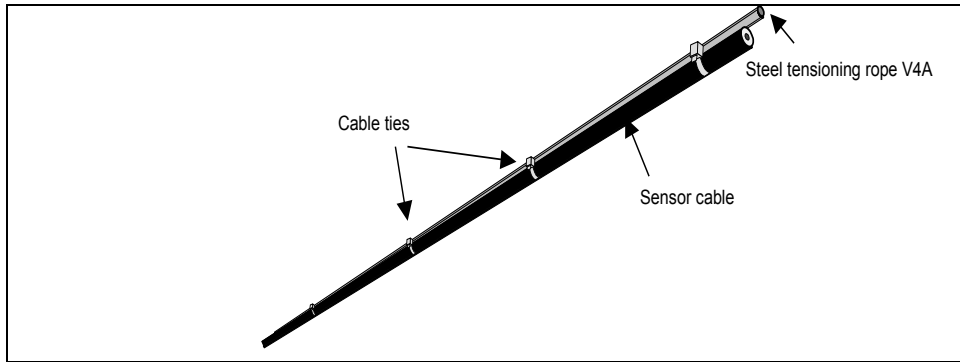


Figure 52: Sensor cable mounted to a tension rope using cable ties

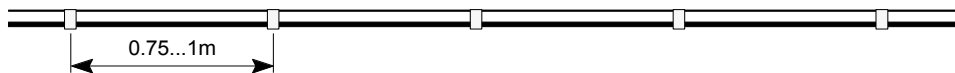


Figure 53: Ties or clamps spacing for mounting in a straight line



Figure 54: Laying at small steps

3.2.11 Fibre-in-Metal-Tube for Power Cables

Sensing fibres in a metal tube (FIMT) are suited for incorporation in the shielding of power cables. A FIMT replaces a shield wire and serves as a distributed monitoring element for power cable shield temperatures.

The optical fibre inside a FIMT has some extra length in order to compensate for thermal expansion of the metal tube. Special care is required to maintain the length of the FIMT precisely during power cable manufacturing. Any change of FIMT length exceeding a few tenth of a ‰ may affect the operation temperature range of a FIMT, and it may even cause extra attenuation. Please ask us for details on proper FIMT handling during cable manufacturing.

Splice and termination boxes with grounding (see section 3.2.7) must be used for FIMT sensor cables.



4 Maintenance and Repair

4.1 Maintenance

The controller is basically maintenance-free. It is also designed to notify automatically any detected failure. Nevertheless, yearly operation tests are required for safety relevant applications. More frequent tests may be considered based on national regulations and risk assessments.

The following procedure should be followed in an operation test:

- Check all sensor cables and pigtails for visible damages such as breaks, cuts, kinks or tight bending
- Switch off the controller and wait for at least 5 seconds
- Switch on the controller and check if all front LEDs lit for a few seconds
- Check if the SYSTEM READY LED lits within 30 seconds
- Connect the controller to a computer and start QUICK START function of the Charon3 software
- Select the controller and scan all channels
- Check the result of the QUICK START for failures and changes compared to the previous QUICK START
- Perform application specific tests as described in the Operation Manuals and local guidelines
- Take appropriate measures in case of failures

4.1.1 Cleaning of Internal Pigtailes

The endfaces of the internal pigtailes of the controller are clean at the time of shipment. They are well protected from contamination by the dummy connectors (Figure 37). Nevertheless, the endface of a pigtail has to be cleaned in case of contamination by a contaminated outer connector or by a missing dummy connector. Contamination of internal pigtailes is indicated by the following:

- the scan of the respective channel within the QuickStart (see Operation Manual) returns the message CHECK CONNECTOR,
- repeated cleaning of the outer pigtail and new scanning does not remove the message and
- the outer pigtail does not show any damage or contamination during inspection.

A Ferrule mate cleaner is the most convenient tool for cleaning an internal pigtail. If no Ferrule mate cleaner is available or its application does not remove the CHECK CONNECTOR message, you have to open the enclosure of the controller.



Warning!

Switch off the controller and disconnect the power supply cable before opening the enclosure. Unscrew the four screws at the optical ports module for at least 1 cm as shown in Figure 55, left. Pull the optical ports module gently partially out of the enclosure (Figure 55, right). Stop pulling when the module is in the position shown in Figure 55 (right) and when you notice a resistance against further pulling. Press the latch of the E2000 connector to be cleaned and gently pull it out of the mating adapter (Figure 55, right, upper connector). Pull the connector completely out of the mating adapter and gently pull the orange or yellow optical fibre cable a few cm out of the enclosure until the connector reaches the position shown in Figure 56, right. Now, clean the connector as described in sec-



tion 3.1.6.4. After cleaning, gently push the connector back into the enclosure without kinking or crushing the orange or yellow optical fibre cable. Carefully insert the E2000 connector into the corresponding inner mating adapter until it securely locks in place. Check the channel number labelled at the optical fibre cable to be identical to that at the front plate. Gently push the optical ports module back into the enclosure without kinking or crushing any orange or yellow optical fibre cable. Tighten the screws.

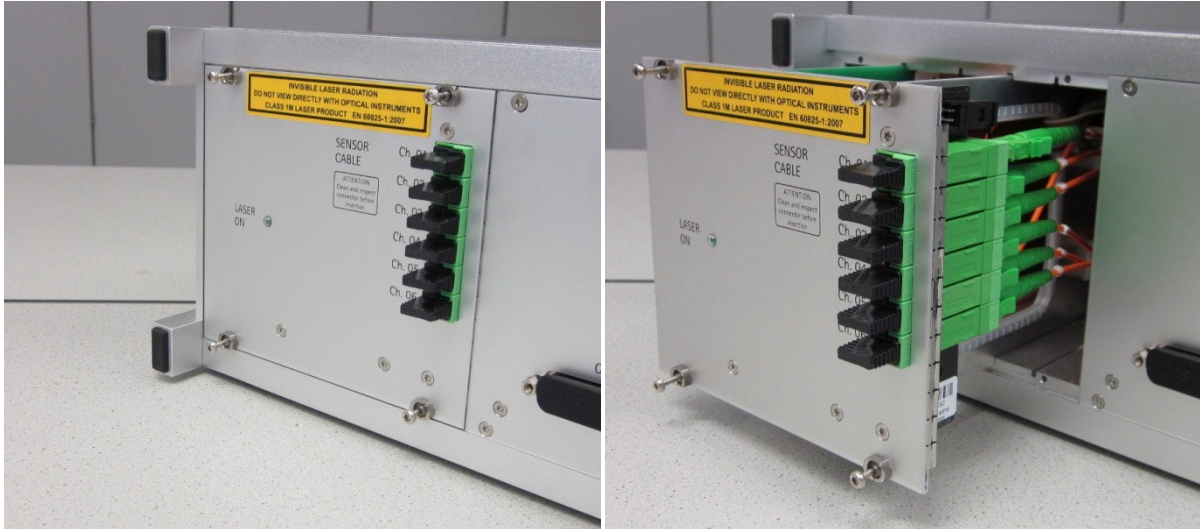


Figure 55: Opening of the optical connector module. Four screws loosened (left). Optical connector module shifted out and internal pigtail of channel 1 partially pulled out of the port (right).



Figure 56: E2000 connector removed from the mating adapter (left) and slightly drawn out of the enclosure (right).

4.2 Repair

The controller must only be repaired by LIOS Technology personnel. Opening the controller as well as breaking any safety seal (Figure 57) lead to a loss of any warranty and product liability.



Figure 57: Safety seal

4.2.1 Exchange of Electrical Fuse

If no LED lights after switching on the controller, please check the supply voltage first. If proper power supply is present, it may be necessary to replace the fuse of the controller. Use fuses of same type as indicated on the controller reverse side only.



Switch off the controller and disconnect the power supply cable before opening the fuse containment at the power supply panel on the reverse side. Exchange the fuse. Close the fuse containment carefully before reconnecting the power cable.

4.2.2 Exchange of Memory Card

The CF memory card (Figure 58) may be exchanged if a memory card with higher capacity is required or if a memory card is faulty. Only cards meeting the specification (see section 5.3) should be used.

Caution!

The EVENT MEMORY and diagnostics data of the controller are saved on the memory card. Therefore, it is essential to save the data. Any warranty and product liability of LIOS Technology is lost if data is not saved before exchange of the memory card and if the instructions given in this section are not followed.

Connect the controller to your PC with Charon3, download the EVENT MEMORY as well as all measurement data and generate a SUPPORT INFORMATION file (see Operation Manual). Send the SUPPORT INFORMATION file to LIOS Support. If the memory card is faulty, it must be sent to LIOS Support.

Switch off the controller and wait for 30 seconds.

Open the MEMORY CARD lid on the rear side (Figure 8) using a Torx T8 screw driver. Break the safety seal in front of the memory card (Figure 59) and press the eject pin at the right (Figure 59, red square) gently using a ball pen or similar device. The memory card unlatches and slides out for a bit. Remove the memory card and insert the new one in the same orientation until it latches. Close the lid. The new memory card will be automatically initialised if necessary.



Figure 58: CF memory card, type I (3.3 mm height)



Figure 59: Open memory card slot with safety seal on the CF memory card and eject pin (marked by the red square)

5 Technical Data

5.1 General Technical Data

Table 19: General technical data

Group	Parameter	Value
Mechanical	Weight, net	13 kg
	Height	131 mm
	Width (without flange)	483 mm (448 mm)
	Depth (without handles)	338 mm (304 mm)
	Rack space	3U, 19-inch wide, 60 cm deep (80 cm with swivelling frame)
Electrical	DC supply voltage, nominal	12 to 48 V ³
	DC supply voltage, max. range	10 to 60 V
	DC start-up voltage, max.	11.5 V
	DC power consumption using 24 V @ 25°C	25 W
	DC power consumption using 24 V @ 60°C	45 W
	DC current, max. @ 12 V	3.5 A
	DC fuse	4 A (T) 5x20 mm
	AC supply voltage, nominal (option) ⁴	100 to 240 V
	AC supply voltage, max. range	85 to 264 V
	AC frequency, nominal	50 to 60 Hz
	AC frequency, max. range	47 to 63 Hz
	AC power consumption using 230 V @ 25°C	35 W / 50 VA
	AC power consumption using 230 V @ 60°C	60 W / 85 VA
	AC current, max. @ 100 V	0.6 A
	AC fuse	2 A (T) 250 V 5x20 mm
	DC supply voltage at AC supply (option), nominal	110 to 220 V
	DC supply voltage at AC supply, max. range	95 to 275 V

³ According to UL/ULC standards, the minimum operating voltage is 14 V.

⁴ The AC supply option has not been evaluated by UL/ULC or VdS.



	DC power consumption at AC supply using 110V	28W @ 25°C, 50W @ 60°C
	Output DC voltage (on INPUT socket)	24 V
	Output DC current	< 20 mA
Laser	Laser class	1M
	Laser wavelength	1550 nm
	Laser power, average	100 mW -EXT3 models: 70 mW -EXT4 models: 16 mW
	Laser power, peak	< 200 mW -EXT3 models: < 150 mW -EXT4 models: < 35 mW
Environmental	Transport and Storage	See Table 20
	Operation	See Table 21
	Protection rating	IP51
Operation	Measuring performance	See datasheets
	Warm-up time	< 1 min + 1 measurement cycle

5.2 Transport and Storage Conditions

The specifications given in Table 20 apply to the transport and storage of the controller in its original transport package (Figure 11, Figure 12). Operation conditions are given in Table 21.

Proper transport, storage, installation and operation conditions are required for a trouble-free and safe operation of the controller. Warranty for the controller is deemed void if these specifications are ignored.

Table 20: Transport and Storage Conditions

Condition	Range
Drop test	< 1 m
Temperature	- 35 to + 75 °C (-31 to 167 °F)
Air pressure (corresponding elevation)	1140 to 540 hPa (- 1000 to + 5000 m)
Relative humidity	10 to 95 % (non-condensing)
Sinusoidal vibration	10-150 Hz: < 20 m/s ²
Shock	< 100 m/s ² , 11 ms

Table 21: Operation conditions

Condition	Range
Temperature	-10 to + 60 °C (14 to 140 °F)
Relative humidity	10 to 95 % (non-condensing)
Air pressure (corresponding elevation)	DC supply: 1140 to 540 hPa (-1000 to 5000 m) AC supply: 1140 to 790 hPa (-1000 to 2000 m)
Sinusoidal vibration (EN 54-22)	10-150 Hz: < 5 m/s ²

5.3 Interfaces

Table 22: Interface specifications

Group	Interface	Parameter	Value
Optical	Optical port(s)	Socket type	E2000/APC (8°)
		Channels	See Table 2



		Fibre type	See Table 16
		Fibre specifications	See Table 18
Power supply	DC supply	Socket type	Sub-D 3WK3-G3
	AC supply (option)	Socket type	IEC320 C14
Data transmission	USB	Socket type	B (USB device)
		Standard	USB 2.0
	Ethernet	Socket type	8P8C (RJ45)
		Number of ports	2
		Data rate, max.	10 (10BASE-T) or 100 MBit/s (100BASE-TX)
	EIA-232 (RS-232)	Socket type	Sub-D 9, male
		Data rate	115.2 or 19.2 kBit/s
		Data/parity/stop bits	8N1
		Handshake	Hardware RTS/CTS
	Inputs	Type	Optocoupler, isolated
		External voltage / current	12-30 V DC / 5 mA
		Socket type	Sub-D 9, female
		Number	4
	Extra inputs (option)	Socket type	Sub-D 25, male
		Number	12, 24 or 36
	Outputs	Type	Photo relay, isolated
		External voltage / current	< 60 V / < 100 mA
		Resistance	< 1.2 Ω
		Socket type	Sub-D 25, female
	Programmable outputs	Number	10
	Collective fault output	Status	Normally closed
	Collective alarm output	Status	Normally open
	Extra outputs (option)	Socket type	Sub-D 25, female
		Maximum number	96
External sensors		Socket type	Sub-D 21W1
(OTS3 option)	Temperature sensors	Type	Pt100 (2 to 4 wires)
		Sensor current	1 mA
		Number of sensors	2
		Measuring range	-200 to 560 °C (300 Ω)
		ADC resolution	22 bit (80 $\mu\Omega$)
		Temperature drift, typical	< +/- 0.05°C
	Current sensors	Current	0 (or 4) to 20 mA
		Input resistance	51 Ω
		Number of sensors	2
		ADC resolution	16 bit (0.3 μA)
		Temperature drift, typical	< +/- 10 μA +/- 0.08%
	Voltage sensors	Voltage	0 to 10 V
		Input resistance	100 k Ω
		Number of sensors	2
		ADC resolution	16 bit (0.15 mV)
		Temperature drift, typical	< +/- 0.05%
	+ 5 V auxiliary voltage	Current, max.	100 mA
	+ 12 V auxiliary voltage	Current, max.	50 mA
	+ 24 V auxiliary voltage	Current, max.	50 mA
Memory card	CF card	Grade	Industrial
		Capacity / GB	2, 8 or 16



5.4 Standards and Approvals

OTS3- and LHD3-series controllers are compliant to international standards and European directives listed in Table 23. The compliance was approved by the notified bodies TÜV Rheinland, VdS Schadenverhütung and Underwriters Laboratories as far as required. Application-specific approvals such as the LHD3-series fire detection approvals are explained in detail in the respective Operation Manual. A full list of applicable safety standards is given in the Declaration of Conformity.

Table 23: Standards and classifications

<i>Requirement</i>	<i>Standard/Directive</i>	<i>Classification</i>	
Electrical safety	IEC 61010-1	CAT II, height 2000 m	
	UL 61010-1		
EMC	IEC 60529	IP 51	
	EN 61326-1:2006	Equipment for measurement, control and laboratory use	
	EN 61000-6-2:2005	Immunity to industrial environment	
	EN 61000-6-3:2007	Emission in residential properties	
	EN 61000-4-2,3,4,5,6,8,11	RF, burst, surge, magn. field, U-var	
	EN 61000-3-2,3	Harmonics and flicker	
	EN 50130-4:1995+A1+A2	Immunity to EMS	
	EN 55011	ISM equipment, Gr. 1, class B	
Laser safety	FCC 47 CFR Ch. 1 Part 15	USA EMI class B	
	IEC 60825-1, -2	Laser class 1M	
Fire detection (LDH3 only)	US 21CFR1040.10		
	FDA accession number	1131367-000	
	pr EN 54-22	Response classes A1N, A2N, BN, CN	
	UL 521	Heat detector also suitable as heat actuated device for special applications	
Environmental testing	ULC-S530		
	IEC 60068-2-6	Sinus vibration 20 m/s ²	
	IEC 60068-2-14	Temperature cycle -10 / + 60 °C	
	NAVMAT P-9492	Random vibration 6.05 g _{rms}	
	ISO 13628-6	Sinus vibration 50 m/s ²	
		Shock 10g, 11ms	
		Temperature -10°C, 60°C	
		Overpressure 199 kPa	
		MIL-STD-810F	Vibration 25.4mm 4.3Hz
		(with IP66 box)	Blowing rain 0.5-4.5mm, wind 18 m/s
(LHD3 only)	FED-STD-101C	Drop test 21" (in transport case)	
	pr EN 54-22	Environmental group II	
RoHS directive	2002/95/EC		
WEEE directive	2002/96/EC		
Low voltage directive	2006/95/EC		
Explosion safety	94/9/EC annex II	II (1) G [Ex op is T3 Ga] IIA	
	EN 60079-0, -28	(-EXT3 models only) II (1) G [Ex op is T4 Ga] IIC and II (1) D [Ex op is Da] IIIC (-EXT4 models only)	



The LHD3 series controllers are approved by UL/ULC and they have been evaluated for open-area protection spacings as outlined in the table 24.

Table 24: Rating of spacing according to UL/ULC

Alarm Settings	Max. spacing	Max. distance from the wall
Fixed Alarm Temperature 135°F (57.2°C) or 190°F (87.8°C) Rate-of-Rise 15°F/min (8.33°C/min) and 20°F/min (11.1°C/min)	50 ft	25 ft
Fixed Alarm Temperature 160°F (71.1°C)	30 ft	15 ft

The fiber optic sensor cable used with the LHD3 series controllers is intended for indoor and outdoor use in a maximum temperature as outlined in table 25.

Table 25: Rating of maximum installation temperature according to UL/ULC

Alarm Settings	Max. installation temperature
Fixed Alarm Temperature 135°F (57.2°C)	100°F (37.8°C)
Fixed Alarm Temperature 160°F (71.1°C)	100°F (37.8°C)
Fixed Alarm Temperature 190°F (87.8°C)	150°F (65.6°C)
Rate-of-Rise 15°F/min and 20°F/min (8.3°C/min and 11.1°C/min)	150°F (65.6°C)

5.5 Hazardous Substances and Recycling

OTS3- and LHD3-series controllers are compliant with the Restriction of Hazardous Substances Directive (RoHS) of the European Community. Exemptions for category 8 and 9 products were not applied in the RoHS assessment.

The controllers comply with the Waste Electrical and Electronic Equipment Directive (WEEE) of the European Community. Discarded controllers shall not be disposed as municipal waste, but collected for recycling together with other electronic and electric equipment.

Each controller contains a Lithium battery containing 0.09 g of Lithium. Discarded batteries shall not be disposed as municipal waste, but collected for proper disposal or recycling together with other batteries.

6 Information Scheme and Contact Information

6.1 Information Scheme

This installation manual forms part of the OTS3 / LHD3 documentation. The following provides an overview of the information scheme for OTS3 / LHD3:

- Datasheets on OTS3 and LHD3 controller models: List the technical data.
- Installation Manual OTS3 / LHD3: Describes the basics of installation of the controller.



- Operation Manual LHD3: Describes basic principles of application of LHD3 controllers in linear heat detection.
- Operation Manual OTS3: Describes basic principles of application of OTS3 controllers in distributed temperature sensing.
- Communication Protocols Manuals: Describes the communication protocols.
 - Part I: LON
 - Part II: DNP3
 - Part III: IEC 60870-5-104
 - Part IV: MODBUS
 - Part V: XML
- Declaration of Conformity: Officially declares compliance with the applicable Directives of the European Commission and the respective European safety standards.

6.2 Contact Information

6.2.1 Online Availability

Updated technical documentation on OTS3 / LHD3 controllers, certificates and software are available for registered customers at the following Internet address:

<https://www.lios-support.com>

Please ask the Technical Support (see section 6.2.4) for registration details.

6.2.2 Representatives and Offices

If you have any further questions relating to products described in this manual, please contact your local representative at the office nearest you.

You can locate your contact partner at this Internet address: <http://www.lios-tech.com>

6.2.3 Training

LIOS Technology offers a variety of training courses to familiarize you with OTS3 / LHD3 controller systems. Please contact your local representative or the Technical Support (see section 6.2.4).

6.2.4 Technical Support

You can reach the Technical Support for all LIOS Technology products using

e mail support@lios-tech.com

Phone +49-221-99887-555 or

FAX +49-221-99887-150.

6.2.5 Contact Information

Company	<i>LIOS Technology GmbH</i>
Department	Sales
Street	Schanzenstrasse 39 Building D9-D13



City	51063 Cologne
Country	Germany
Phone	+49 221 99887 0
FAX	+49 221 99887 150
e-mail	info@lios-tech.com
Internet URL	http://www.lios-tech.com



Figures

Figure 1: Warning label	7
Figure 2: Examples of the nameplates of OTS3-series and LHD3-series controllers.....	10
Figure 3: System architecture	11
Figure 4: Front view of an OTS3-series controller.....	12
Figure 5: Front view of a 4-channel LHD3-series controller.....	13
Figure 6: Rear view of a controller with optical ports, electrical interfaces and power supply panels	14
Figure 7: Optical ports panels of 2, 8 and 16 channel controllers.....	15
Figure 8: Standard interface panel.....	15
Figure 9: Extended interface panel	15
Figure 10: DC (left) and AC (right) power supply panels	16
Figure 11: Standard transport box	20
Figure 12: Optional heavy duty transport case	20
Figure 13: Shock label. A red tube in the centre would indicate rough handling.....	21
Figure 14: 19-inch rack comprising controller, computer and UPS	23
Figure 15: Left and right screws locations.....	23
Figure 16: Power Supply Cable DC 12-48V.....	24
Figure 17: Power Supply Cable 230V AC, European Plug.....	25
Figure 18: LIOS Ethernet patch cable	25
Figure 19: LIOS cross-over Ethernet cable	26
Figure 20: USB cable.....	26
Figure 21: Serial interface cable	27
Figure 22: Output cable.....	28
Figure 23: Example for connecting an alarm control system to outputs of the controller.....	29
Figure 24: Input Cable (4 Inputs, 24V DC Output).....	30
Figure 25: Examples for connecting fire detectors to inputs of the controller.....	31
Figure 26: Input Cable (12 Inputs).....	32
Figure 27: External sensors cable.....	33
Figure 28: Connection schemes for a Pt100 temperature sensor in 2-, 3- and 4-wire configurations.	34
Figure 29: Connection scheme for a 0 to 20 mA current sensor using a supply voltage of 24V.....	34
Figure 30: Connection scheme for a 4 to 20 mA current sensor using a supply voltage of 24 V.....	35
Figure 31: Connection scheme for a 0 to 10 V voltage sensor using a supply voltage of 24 V.....	35
Figure 32: Pigtail with E2000/APC connector.....	36
Figure 33: Mounting of splice boxes inside a rack	37
Figure 34: Fibre scope with 200x magnification, built-in laser filter and LED illumination	37
Figure 35: Views of clean (left), contaminated (middle) and damaged (right) pigtail endfaces.....	38
Figure 36: Cleaning of a ferrule endface of an E2000/APC connector using an optical wipe.....	38
Figure 37: Dummy connector.....	38
Figure 38: Ferrule mate cleaner	39
Figure 39: Sensor cable Metal Free FRNC	40
Figure 40: Sensor cable Fast Response FRNC.....	40
Figure 41: Splice cassette	44
Figure 42: Splice box without grounding and splice cassette	44
Figure 43: Splice box with grounding and splice cassette.....	45
Figure 44: Splice box with cable and pigtail	46
Figure 45: Termination box	46



Figure 46: Termination kit comprising hot-melt splice-protection sleeves and black heat-shrinkable tubings..... 47

Figure 47: Cable laying for repair 48

Figure 48: Sensor cable mounted with cable ties 48

Figure 49: Nail plug pin..... 49

Figure 50: Cable clip 49

Figure 51: Tube clamp for fixing larger diameter sensor cables 49

Figure 52: Sensor cable mounted to a tension rope using cable ties 50

Figure 53: Ties or clamps spacing for mounting in a straight line..... 50

Figure 54: Laying at small steps..... 50

Figure 55: Opening of the optical connector module. Four screws loosened (left). Optical connector module shifted out and internal pigtail of channel 1 partially pulled out of the port (right)..... 52

Figure 56: E2000 connector removed from the mating adapter (left) and slightly drawn out of the enclosure (right)..... 52

Figure 57: Safety seal..... 53

Figure 58: CF memory card, type I (3.3 mm height)..... 54

Figure 59: Open memory card slot with safety seal on the CF memory card and eject pin (marked by the red square)..... 54



Tables

Table 1: Safety symbols and notices	5
Table 2: Controller models	10
Table 3: Front LEDs of OTS3-series a controller	12
Table 4: Front panel LEDs of a LHD3-series controller	13
Table 5: Functions of the ACKNOWLEDGEMENT key switch.....	14
Table 6: Rear panel LEDs of the controller	14
Table 7: Buzzer beep code for different events	16
Table 8: Null modem cable cross-over linking scheme	26
Table 9: Pin and lead colour assignment of outputs	27
Table 10: Pin assignment of extra outputs.....	29
Table 11: Charon3 output number range for each output socket	29
Table 12: Pin and lead colour assignment of standard inputs	30
Table 13: Charon3 input number range for each input socket	31
Table 14: Pin and lead colour assignment of extra inputs	32
Table 15: Pin and lead colour assignment of external sensors	32
Table 16: Fibre types	35
Table 17: Examples for sensor cables.....	39
Table 18: OTS3 fibre and sensor cable specifications	41
Table 19: General technical data.....	54
Table 20: Transport and Storage Conditions.....	55
Table 21: Operation conditions	55
Table 22: Interface specifications.....	55
Table 23: Standards and classifications.....	57
Table 24: Rating of spacing according to UL/ULC.....	58



Abbreviations

AC	Alternating current
ADC	Analogue digital converter
APC	Angled physical contact
CD	Compact disk
CF	Compact Flash
CFR	Code of Federal Regulations
CTS	Clear to send
DC	Direct current
DCD	Data carrier detect
DHCP	Dynamic host configuration protocol
DNS	Domain name system
DSR	Data set ready
DTR	Data terminal ready
DTS	Distributed temperature sensing
EC	European Commission
EMC	Electromagnetic compatibility
EN	European standard
ESD	Electrostatic discharge
EX	Explosion (protection)
FIMT	Fibres in metal tube
FRNC	Flame retardant non-corrosive
GI	Graded index
HU	Height units
IEC	International Electronic Commission
IP	International protection rating / Internet protocol
LAN	Local area network
Laser	Light amplification by stimulated emission of radiation
LED	Light emitting diode
LHD	Linear heat detection
MEMS	Micro-electromechanical system
MM	Multimode
N.A.	Numerical aperture
NTP	Network time protocol
OTS	Optical temperature sensing
PA	Polyamide
PC	Personal computer
PDF	Portable data format
RAM	Random access memory
RI	Ring indicator
RoHS	Restriction of hazardous substances
RTS	Request to send
RTTR	Real-time thermal rating
RxD	Receive data
SAGD	Steam assisted gravity drainage
SCADA	Supervisory control and data acquisition
SM	Single-mode
TCP	Transmission control protocol
TxD	Transmit data
UL	Underwriters Laboratory
USB	Universal serial bus
UPS	Uninterruptible power supply



UV	Ultraviolet
VGA	Video graphics array
WEEE	Waste electrical and electronic equipment

Glossary

<i>Acknowledge</i>	Acknowledging an alarm confirms that you have noted it. The buzzer is deactivated by the acknowledgment.
<i>Backscattering curve</i>	Set of anti-Stokes or Stokes signal power data in dependence on distance.
<i>Controller</i>	General term for LIOS distributed sensing devices.
<i>Cycle time</i>	Time for measuring a complete sequence of channels.
<i>Distance range</i>	Maximum length of optical fibre to be measured.
<i>EMC</i>	Electromagnetic compatibility is the ability of electrical equipment to function properly in its electromagnetic environment without influencing this environment.
<i>Measuring time</i>	Time required to measure a temperature profile along an optical fibre.
<i>Output, manual reset</i>	A manual reset output does not automatically deactivate when the activating condition is no longer present. Deactivation requires a reset of the controller. Manual reset outputs may be used for alarm and fault indication.
<i>Output, self-reset</i>	A self-reset output is automatically deactivated when the activating condition is no longer present. Self-reset outputs may be used to control other devices.
<i>Pigtail</i>	A pigtail is a short optical fibre cable with an optical connector on one end and a length of exposed optical fibre at the other end.
<i>Ping</i>	Ping is a computer network utility for testing a connection between computers on an Internet protocol (IP) network.
<i>Reset</i>	Resetting the controller confirms that events triggering alarms and failures have been properly handled. All alarms, failure states and manual reset outputs are reset.
<i>Return loss</i>	In fibre optics, return loss is the loss of signal power due to a reflection, e.g. caused by a discontinuity of refractive index at a connector, another optical component or the fibre endface.
<i>Sampling interval</i>	Distance between measuring data locations along a fibre.
<i>Spatial resolution</i>	Length of a hot spot required for 90% temperature reading response. When the length of a hot spot is shorter than the spatial resolution, the temperature reading is lower by a factor of approximately the ratio of hot spot length and spatial resolution.
<i>Temperature accuracy</i>	Deviation between average measured temperature and true temperature.
<i>Temperature profile</i>	Set of temperature data in dependence on distance.
<i>Temperature resolution</i>	Standard deviation of local temperature data taken in a measuring sequence.
<i>Warm-up time</i>	Time between power-on and readiness to take measurements according to the specification.







This Page Left Intentionally Blank

OPERATION MANUAL

LHD3- AND LHD3S SERIES CONTROLLERS

Introduction

System Introduction

Software Settings

Commissioning

Operating the Controller

Troubleshooting

Contact Information



Table of Contents

- Table of Contents 2
- 1 Introduction 6
 - 1.1 General Information 6
 - 1.1.1 Purpose of the Operation Manual..... 6
 - 1.1.2 Qualified Personnel 6
 - 1.2 Legal Notices..... 6
 - 1.3 Applications 6
 - 1.3.1 Fire Detection in Tunnels..... 6
 - 1.3.2 Fire Detection in Parking Decks..... 7
 - 1.3.3 Fire Detection in Buildings..... 7
 - 1.3.4 Fire Detection in Cable Trays..... 7
 - 1.3.5 Fire Detection in Industrial Equipment 7
 - 1.3.6 Fire Detection at Pipelines and Tanks 7
- 2 System Introduction..... 8
 - 2.1 Function..... 8
 - 2.2 Locations and Zones 8
 - 2.3 Triggering Parameters 9
 - 2.4 Fire Characteristics 12
 - 2.5 Acknowledgement and Reset..... 12
 - 2.6 Alarm 13
 - 2.7 Pre-Alarm..... 13
 - 2.8 System Fault 13
 - 2.9 Fibre Break..... 13
 - 2.10 Test Mode..... 14
 - 2.11 Inputs..... 14
 - 2.12 Outputs..... 14
 - 2.13 Address Points 15
- 3 Software Installation 16
 - 3.1 General Procedure 16
 - 3.2 Setup..... 16
 - 3.3 Starting Charon3 and Setting of User Levels 22
 - 3.4 Software Update 24
 - 3.5 USB Driver Installation..... 25



4	Commissioning the Controller	28
4.1	Quick Start	28
4.2	Trigger Parameters	35
4.2.1	Alarm Parameter Sets.....	35
4.2.2	Control Parameter Sets	38
4.3	Communication Settings	38
4.3.1	Event Configuration.....	39
4.3.2	Network Settings	44
4.3.3	LON Protocol Settings.....	44
4.3.4	Other Protocols Settings.....	46
4.3.5	Locking the Controller	49
4.4	Measurement Sequence	50
4.5	Fibre Settings.....	51
4.5.1	Measurement Parameters	51
4.5.2	Temperature Calibration	52
4.5.3	Zone Configuration.....	53
4.5.4	Fibre Events	55
4.5.5	Length Scaling.....	56
4.6	Commissioning Using Different Charon3 Instances	57
5	Operating the Controller.....	59
5.1	Measurement	59
5.2	Single Channel Measurement	59
5.3	Acknowledgement and Reset.....	59
5.4	Evaluation of Sensor Fibres Using Backscattering Profiles.....	60
5.4.1	Attenuation	60
5.4.2	Point losses.....	61
5.4.3	Termination	61
6	Servicing the Controller	63
6.1	Data Recording	63
6.1.1	Recording Settings.....	63
6.1.2	Data Download	64
6.2	Test Mode.....	65
6.2.1	Test Mode Settings.....	65
6.2.2	Activation and Deactivation of the Test Mode	66
6.2.3	Performing Controller Tests	67



6.3	Support	68
6.3.1	Event Memory	68
6.3.2	Configuration Report	69
6.3.3	Status Report	70
6.3.4	Project Information	71
6.3.5	Support Information	71
6.3.6	Firmware Update	74
6.4	Optional Licenses	76
7	Data Visualization	76
7.1	Quick Access Toolbar	76
7.2	Edit	78
7.3	View	78
7.3.1	Tools	78
7.3.2	Options	80
7.4	Windows	89
7.4.1	Temperature Profile	90
7.4.2	Backscattering Profile	92
7.4.3	Zone Data	92
8	Data Handling	94
8.1	Import	94
8.2	Export	94
9	Parameter Sets	98
9.1	EN 54-22 Approved Parameter Sets	98
9.2	UL521 / ULC S530 Approved Parameter Sets	99
9.3	Other Parameter Sets	100
9.4	Parameter Sets for EX Controllers	101
10	Computer Requirements and Software License	103
10.1	Computer Requirements	103
10.1.1	Hardware Requirements	103
10.1.2	Operating System	103
10.2	Software License	104
10.2.1	Subject of the Contract	104
10.2.2	Limited Warranty	104
10.2.3	Liability	105



10.2.4	Safeguard against Misuse.....	105
10.2.5	Open Source Licenses.....	105
10.2.6	Other Restrictions.....	105
10.2.7	Governing Law.....	105
10.2.8	No Assignment.....	105
11	Troubleshooting.....	106
11.1	General.....	106
11.2	Fibre Address Points.....	106
11.3	Errors.....	106
11.4	Warnings.....	107
11.5	Messages.....	107
11.6	Network Ports.....	108
12	Information Scheme and Contact Information.....	109
12.1	Information Scheme.....	109
12.2	Contact Information.....	109
12.2.1	Online Availability.....	109
12.2.2	Representatives and Offices.....	109
12.2.3	Training.....	109
12.2.4	Technical Support.....	109
12.2.5	Contact Information.....	110
	Figures.....	111
	Tables.....	114
	Abbreviations.....	115
	Glossary.....	116



1 Introduction

1.1 General Information

1.1.1 Purpose of the Operation Manual

This Operation Manual provides information based on the requirements for engineering documentation. The information relates to the LHD3-series controllers, its commissioning and operation. The instructions given in this Operation Manual presume that all controller hardware is properly installed and maintained according to the Installation Manual for OTS3- / LHD3-series controllers. The instructions are intended for:

- Commissioning engineers
- Operators

General knowledge of software, optical sensing, and fire detection is needed to understand the commissioning and operating instructions. It is also assumed that those using the Operation Manual have experience in using personal computers running Windows operating systems.

1.1.2 Qualified Personnel

The controller can only be used in conjunction with this documentation. Commissioning and operation of the controller can only be performed by qualified personnel. Within the context of this documentation qualified persons are persons, who are authorized to commission and operate devices, systems and circuits in accordance with established safety practices and standards.

1.2 Legal Notices

We have reviewed the contents of this publication to ensure consistency with the hardware and software described herein. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions. The original language version of this publication is German.

It is our policy to constantly improve the design and specifications. Accordingly, the details represented in this manual cannot be regarded as final and binding.

The contents of this manual, in particular all texts, pictures, graphics, and logos are subject to the copyright of LIOS Technology GmbH. Any printing, copying, publishing, presentation, translation or modification of the manual or parts of its contents requires a written permission from LIOS Technology GmbH.

1.3 Applications

The LHD3-series controllers are approved as line type heat detector for fire detection. Their approvals base on UL/ULC and EN standards. Accordingly the LHD3-series controllers were only evaluated for maximum temperature and rate of rise alarm.

1.3.1 Fire Detection in Tunnels

LHD3-series controllers are approved linear heat detectors for fire detection in tunnels and buildings. They are also approved for fire detection at equipment operated at elevated temperatures. Typical measuring times last a few seconds only. A fire alarm is actuated within one to a few minutes depending on temperature rates of rise and in compliance with the respective national standards.



Maximum temperature and local differential criteria also serve as triggers for fire alarm actuation. Besides alarm actuation, the controller also evaluates and indicates the location, size and propagation direction of a fire. Details are given in the national fire detection approvals of the controller.

Typically, a temperature reading is taken every 3 m for long objects such as road, cable and railway tunnels.

1.3.2 Fire Detection in Parking Decks

For fire detection in parking decks, temperature readings are typically taken every 1 m.

1.3.3 Fire Detection in Buildings

For fire detection in buildings and other smaller objects, temperature readings are typically taken every 0.5 m. Optional smaller temperature sampling intervals can be used in special applications requiring the detection of small-sized fires, e.g. at individual seats in a concert hall.

1.3.4 Fire Detection in Cable Trays

Distributed temperature monitoring along a power cable using an LHD3-series controller enables safe cable operation at any location even at high loads. The optical fibre sensor cable can be integrated into the shielding of the power cable or attached externally. It is immune to electromagnetic interference.

1.3.5 Fire Detection in Industrial Equipment

Temperature monitoring is essential in a number of industrial processes. Typical application temperatures are quite high, and high spatial as well as temporal resolutions are required. LHD3-series controllers provide excellent spatial resolution for such applications and shortest measurement times. Our sensor cables for industrial applications are robust and suited for high temperatures (see Installation Manual).

Further examples of LHD3-series controller's applications in industry are chemical plants, baking of composite materials, gas turbines, power plants and different steps of metal processing. Immunity against electromagnetic interference and excellent chemical stability of the optical fibre sensor cable are clear advantages of optical sensing compared to electrical measurements in a number of applications.

1.3.6 Fire Detection at Pipelines and Tanks

At pipelines and tanks, controllers of the LHD3-series may be used for a precise and fast fire detection.



2 System Introduction

2.1 Function

The controller takes spatially resolved temperature profiles along optical fibre sensor cables. The controller software evaluates temperature profiles as well as their temporal and local variations according to maximum or minimum temperature, temperature slope and local differential criteria in order to trigger fire alarms, pre-alarms or control systems.

The controller is intended for stand-alone operation without a permanent link to a computer. Computers can be temporarily or permanently connected for commissioning, control and data download. Alarm control systems can be “hard-wired” connected to the in- and output relays of the controller. Further third party systems (e.g. SCADA systems) can also be connected to the serial and Ethernet interfaces using the standard (LON, XML) and optional protocols.

2.2 Locations and Zones

The controller analyses temperatures at hundreds to thousands of locations along the optical fibre sensor cable. The spacing between these locations can be 0.5, 1 or 3 m according to the controller settings. In most fire detection applications, the supervised area is split into a number of sections such as rooms, sprinkler zones or other fire detection zones. The controller software can group the locations along an optical fibre sensor cable in a similar way into up to 1000 zones (Figure 1) with different distance ranges defined by the user (see section 4.5.3). The user can select sets of triggering parameters for each of the zones. Temperatures at all locations within a zone are then evaluated according to the triggering parameter sets of the zone.

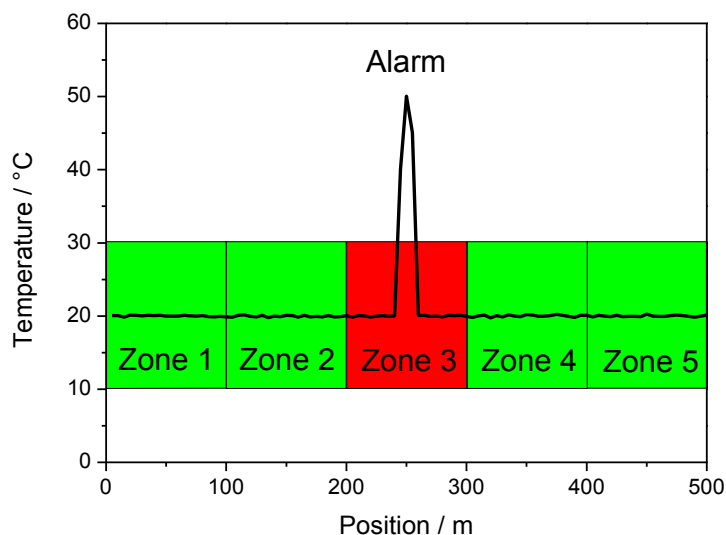


Figure 1: Fibre length of 500 m divided into 5 zones. An alarm is triggered in zone 3 by elevated temperatures around 250 m.

In case of a fire or another event matching the triggering parameters, the respective zone triggers an event such as an alarm, pre-alarm or control. Besides optical channel and zone numbers, the following additional information on the alarm triggering event can be transmitted by the controller:

- Triggering location



- Triggering parameter
- Size of fire
- Propagation direction of fire

2.3 Triggering Parameters

The controller can save up to 32 different sets of triggering parameters in the system memory. Each set is assigned to one of the following two groups:

- Alarm parameter set used for triggering alarms, pre-alarms or test alarms
- Control parameter set used for controlling other equipment, e.g. ice warning indicators, air ventilation, ...

Alarm parameter sets can include the following parameters:

- Maximum temperature within the zone
- Minimum temperature within the zone
- Hot spot temperature difference (temperature difference between the highest and the average temperature within the zone)
- Cold Spot temperature difference (temperature difference between the lowest and the average temperature within the zone)
- Up to three different rates-of-rise (temperature gradients) within the zone
- Alarm Delay

The LHD3-series controllers were only evaluated according to UL/ULC and VdS standards for maximum temperature and rate of rise alarms.

Alarm triggering at a maximum temperature T_{\max} (Figure 2) is a standard feature of fire detection systems. It reliably triggers an alarm or pre-alarm in case of slow heating over hours, such as it is the case for smouldering fires. The minimum temperature parameter can be used as an additional feature for alarm actuation in applications requiring e.g. leak detection or ice warning.

The hot-spot temperature difference (Figure 3) is especially useful for early detection of localized heating and in situations where the sensor cable is only slightly heated. The alarm is triggered when the temperature difference between the highest temperature in a zone and the zone average temperature exceeds the set value for hot spot ΔT . For instance, overheating of a bearing of a conveyor belt is restricted to a small location and the sensor cable can be installed in a certain distance from the bearing. Then, the cable experiences just a small, localized heating. The maximum temperature parameter can not be suited for alarm actuation since the cable heating is within the range of normal environmental temperature variations.

The cold spot parameter can be used for alarm triggering in leak detection.

The rate-of-rise parameters (Figure 4) are especially useful for the early detection of fast starting fires. The alarm is triggered when the temperature difference ΔT in a time interval Δt exceeds the set value. The rate-of-rise parameters can trigger an alarm before the maximum permitted temperature is reached. The fast alarm triggering in case of temperature rise is also required in most national standards on fire detection.



The alarm delay parameter determines how many subsequent alarm triggering events are required to trigger an alarm. If no alarm delay is set, every event immediately triggers an alarm. If the alarm delay is set to 1, an alarm in a zone is only triggered if in both of two subsequent measurements at least one alarm triggering event is detected within the zone.

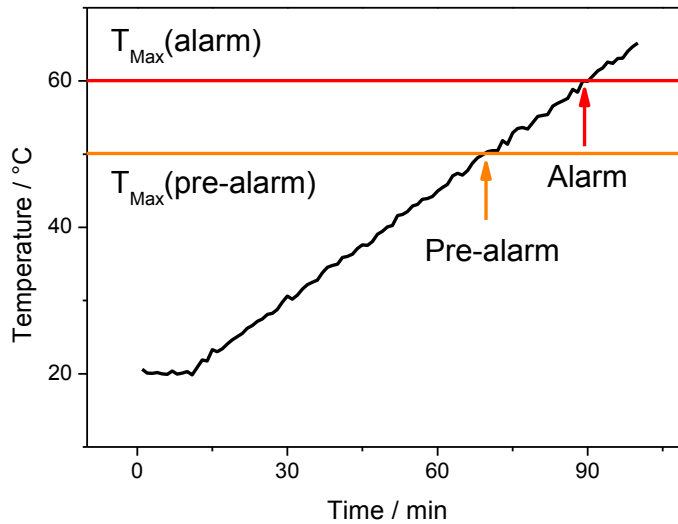


Figure 2: Pre-alarm and alarm triggering at maximum temperatures

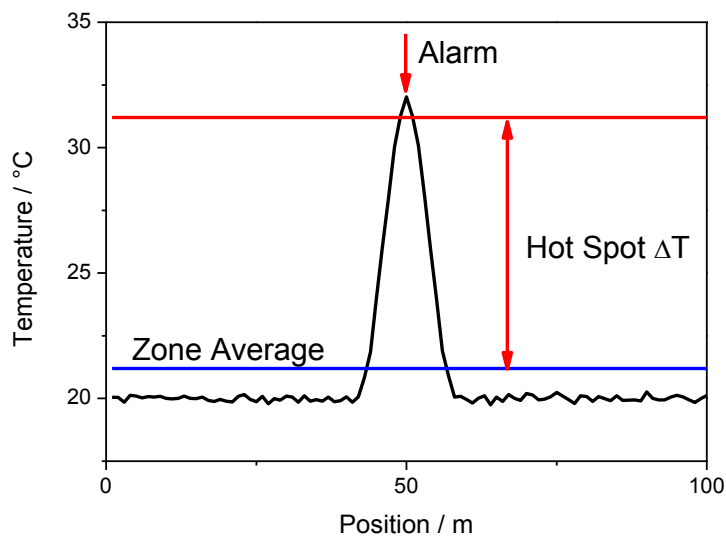


Figure 3: Alarm triggering at a hot spot temperature difference

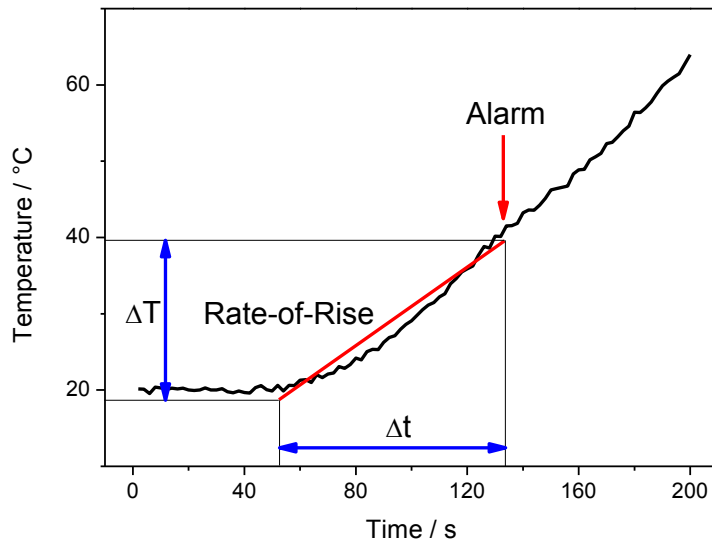


Figure 4: Alarm triggering at a rate-of-rise

A control parameter set includes the parameters:

- Activation temperature
- Deactivation temperature

A two-step control can be realized using the two switch temperature parameters (Figure 5). An ice warning indicator can be controlled using an activation temperature lower than the deactivation temperature. The indicator is then activated when at least one location within the zone has a lower temperature than the activation temperature. The indicator is deactivated when all locations within the zone have higher temperatures than the deactivation temperature. Air ventilation can be controlled in a similar way, but normally with a higher activation temperature than the deactivation temperature (Figure 5).

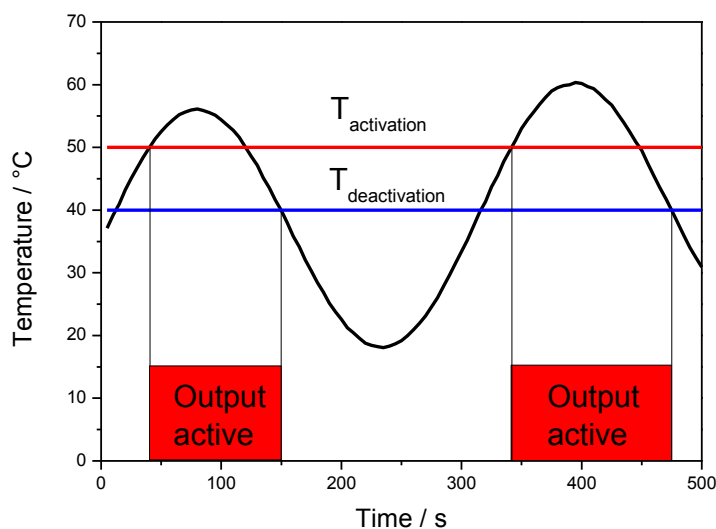


Figure 5: Two-step control of an output using an activation temperature of 50°C and a deactivation temperature of 40°C



2.4 Fire Characteristics

The controller evaluates size class and propagation direction of a fire. The fire magnitude is defined by the distance range of adjacent alarm triggering locations. Each fire is classified into one of up to five size classes (Table 1). The maximum size of each class can be configured (see section 4.5.4).

Table 1: Fire magnitude classes

Fire magnitude class	Maximum size (default values)
1	3 m
2	10 m
3	30 m
4	100 m
5	300 m

The controller analyses the propagation direction of a fire by evaluating the movement of the centre point of the fire. Fire propagation directions are given in Table 2.

Table 2: Fire propagation directions

Fire propagation direction	Characteristics
Static	No significant movement
Towards fibre end	Centre point moves towards fibre end
Towards controller	Centre point moves towards controller

The evaluation of fire characteristics was not evaluated by UL/ULC and VdS.

2.5 Acknowledgement and Reset

The deactivation of alarms, pre-alarms and system faults requires an ACKNOWLEDGEMENT and/or a RESET.

ACKNOWLEDGEMENT and RESET can be triggered by:

- Key switch actuation
- Input activation
- Commands from Charon3 or third party software

The RESET is also triggered by:

- Starting a continuous measurement
- Switching the controller off and on
- Activating / deactivating the test mode

An ACKNOWLEDGEMENT deactivates the buzzer. A RESET deactivates the buzzer, alarm and fault LEDs as well as the non-self-resetting and the collective outputs. The RESET also deactivates the self-resetting outputs and the pre-alarm if these options are activated (section 4.3.1). Furthermore, the configured system address points are transmitted after both, ACKNOWLEDGEMENT and RESET.



In the test mode (section 6.2), a FULL RESET command is available in order to reset all events, inputs and outputs. The FULL RESET acts like a RESET plus a deactivation of inputs and self-reset outputs.

2.6 Alarm

An alarm is an event triggered when temperatures or temperature trends within a zone exceed the limits of the alarm parameter set of that zone. The alarm is indicated by the red ALARM LED, the buzzer, the collective alarm output and optionally programmable outputs according to the zone configurations. An alarm message and an address point according to the zone configuration are transmitted using the communication protocol(s). The buzzer is deactivated by an ACKNOWLEDGEMENT, and the alarm is deactivated by a RESET. A blinking ALARM LED indicates an alarm being no more active, but not yet reset.

2.7 Pre-Alarm

A pre-alarm is an event comparable to an alarm, but triggered using the pre-alarm parameter set of a zone and indicated by the PRE-ALARM LED instead of the ALARM LED. The collective alarm output is not activated by a pre-alarm. According to the controller settings (see section 4.3.1), a pre-alarm can be deactivated automatically if the triggering condition is no longer fulfilled, or a reset can be required for deactivation. A pre-alarm is not to be used for evacuation purposes.

2.8 System Fault

A system fault is an event triggered by one of the following conditions:

- Controller fault
- Fibre break
- Power off
- No active measurement
- Environmental conditions (supply voltage, operating temperature, moisture) out-of-range
- User fault

A system fault is indicated by the yellow SYSTEM FAULT LED and the collective fault output. A fibre break is also indicated by the FIBRE FAULT (FibroLaser: FAULT) LED and optionally programmable outputs according to the zone configurations.

A fault message and an address point according to the controller and zone configurations are transmitted using the communication protocol(s). The system fault is deactivated by a RESET.

2.9 Fibre Break

The controller takes measurement data along an optical fibre sensor cable(s). Although the optical fibre is well protected by the sensor cable, it may break if the sensor cable is cut, tightly bent, crushed, impacted or subjected to high tensile forces. The controller continuously monitors the fibre length, and it detects a fibre break in case of any clear reduction of the length. The fibre break triggers a system fault event (section 2.8). The controller can be configured to continue or to stop measurements in case of a fibre break. The controller can detect heat using the broken fibre, but only to a position 10 sampling intervals (i.e. 30 m for 3 m sampling interval) ahead of the fibre break. The sensor cable must be repaired in order to monitor the full length again (see Installation Manual).



2.10 Test Mode

The controller can be operated in a test mode for commissioning and maintenance purposes. The test mode is activated using a correspondingly configured input or a command from Charon3 (see section 6.2.2) or a third party software. Transmission of alarms and other signals to fire control panels and other third party systems can be deactivated in test mode (see section 6.2.1). A special alarm parameter set can be used in test mode instead of normal alarm parameter sets.

2.11 Inputs

Digital inputs of the controller can be used to trigger alarms, controls, acknowledgement or reset. The active level of any input (high or low), its mode (alarm, control, acknowledge, deactivate buzzer, buzzer/LED test, test mode or reset) and optionally related outputs can be assigned individually (see section 4.3.1). An input operated in the alarm mode can be used to integrate point type fire detectors into the alarm triggering scheme of the controller.

2.12 Outputs

The controller provides two predefined outputs for collective alarm and system fault as well as 10 (optionally more) programmable outputs (see Installation Manual).

The collective alarm output is normally open, whereas the system fault output is normally closed (see Installation Manual).

The collective alarm output is activated (closed) by any alarm within one of the zones at any measured fibre. The system fault output is activated (opened) by any event given in section 2.8.

In case of a power down, all outputs are open. This indicates a system fault, but no collective alarm.

Programmable outputs can be assigned to

- Alarms
- Pre-alarms
- Controls
- Inputs
- Faults
- Warnings

The active state of any programmable output (normally open or normally closed) and its function can be assigned individually (see section 4.3.1).

The switching characteristics of outputs can be MANUAL RESET or SELF-RESET. A manual reset output requires a reset for deactivation, whereas a self-reset output deactivates automatically if the triggering condition is no longer fulfilled. Alarm and fault outputs are manual reset, and control outputs are self-reset. The characteristics of pre-alarm outputs depend on the controller settings (section 4.3.1). The characteristics of outputs assigned to inputs are determined by the input mode (section 4.3.1). Charon3 ensures that all events assigned to an output have the same switching characteristics.

If several triggering conditions are used for a self-reset output, the output is activated if at least one of the conditions is fulfilled (activated), and it is deactivated if all conditions are not fulfilled (deactivated).



2.13 Address Points

An ADDRESS POINT is the identification number of an event. Address points are used for notifying events to Charon3 and to third party systems using telegrams of communication protocols. Address points can be assigned to the following events:

- Alarms
- Pre-alarms
- Controls
- Inputs
- Faults
- Warnings
- General (reset, acknowledgement, test mode)

There are two groups of address points: fibre address points assigned to zone events and other fibre events and system address points assigned to events not related to fibres or zones.

Fibre address points are assigned to alarms, pre-alarms, controls, fibre breaks and other fibre events (fire magnitude and direction) according to a fixed scheme (see section 11.2).

System address points have a default assignment to events (see section 11.2). The assignment of system address points can be modified by the user if required (see section 4.3.1).



3 Software Installation

3.1 General Procedure

The Charon3 software needs to be installed on a suitable computer (see section 10.1) using the Charon3 software CD. Install the software only after reading and accepting the license (see section: 10.2). Please use the following general procedure for installation:

- Log in using an user account with administrative privileges.
- Do not run any other application. Quit all programs run by AUTOSTART.
- Insert the Charon3 CD.
- The SETUP program runs automatically if the AutoRun feature is activated on your computer. If the SETUP program does not automatically run, execute the SETUP.EXE application in the Charon3 folder on the CD.
- Follow the instructions displayed by the SETUP wizard (see section 3.2).
- Click the Charon3 entry in the start menu to run Charon3. Charon3 generates a database during first execution. This process can take a few minutes.

3.2 Setup

The window shown in Figure 6 indicates that the SETUP has been started.

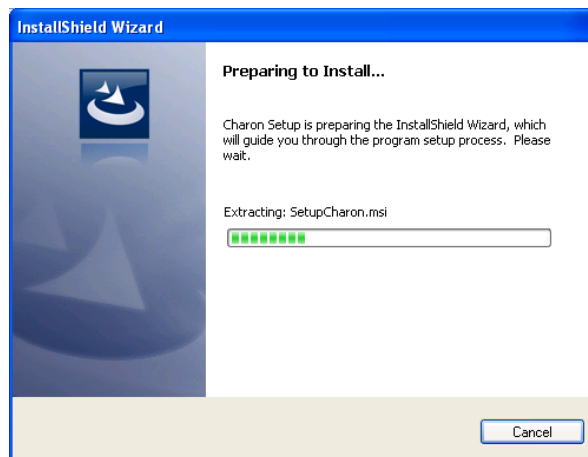


Figure 6: Preparing to install message

After a few seconds, the welcome window of the InstallShield Wizard (Figure 7) opens. Each window of the InstallShield Wizard offers options to continue installation (Next), to go back to the previous step (Back) or to cancel the installation (Cancel). The InstallShield Wizard automatically continues the installation unless you press the CANCEL button. Then, the license agreement window (Figure 8) opens¹. You must read and accept the agreement to continue installation.

¹ The full text of the license agreement is also shown in chapter 10.2 of this manual.

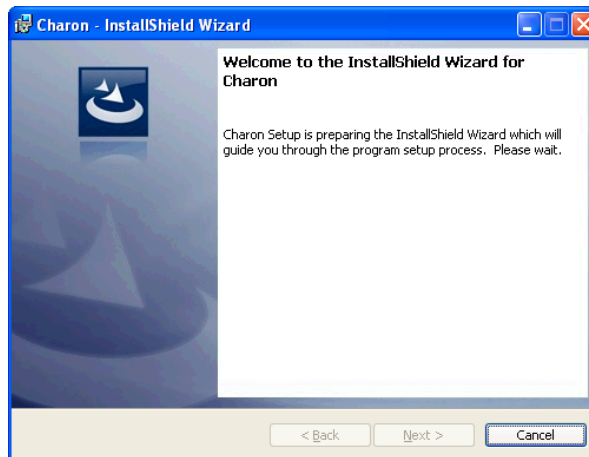


Figure 7: Welcome window of the InstallShield Wizard



Figure 8: License agreement window

The next window of the wizard (Figure 9) offers the option to change the folder of the Charon3 installation. Select NEXT for an installation in the standard path c:\Program Files\LIOS\. You can select CHANGE if you want to install Charon3 in a different folder.

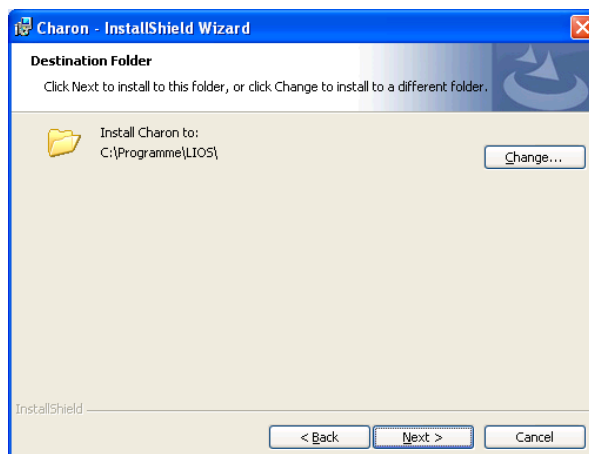


Figure 9: Selection of destination folder

The next window of the wizard (Figure 10) offers the option to change the folder of the database installation. Select NEXT for an installation in the standard path c:\OTSDData. You can select CHANGE



if you want to install the database in a different folder. Select a drive with sufficient free space and read/write access.

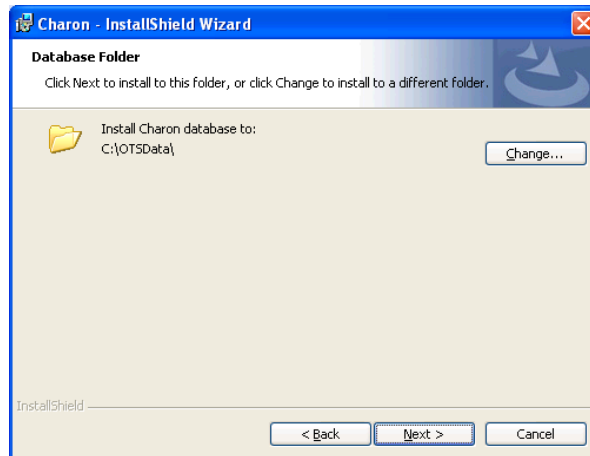


Figure 10: Selection of the database folder

Charon3 is now ready for installation. Press the **INSTALL** button in the Ready-to-Install window (Figure 11) to start the installation on your computer. The installation can take several minutes depending on the performance of your computer.

If another installation (e.g. a Windows update) is running at the same time, the error message shown in Figure 12 is displayed. Wait until the other installation is finished and press **RETRY** or press **CANCEL** to cancel the Charon3 installation. Confirm the cancellation by pressing **YES** in the confirmation window (Figure 13). Repeat the Charon3 installation after completion or cancellation of the other installation.

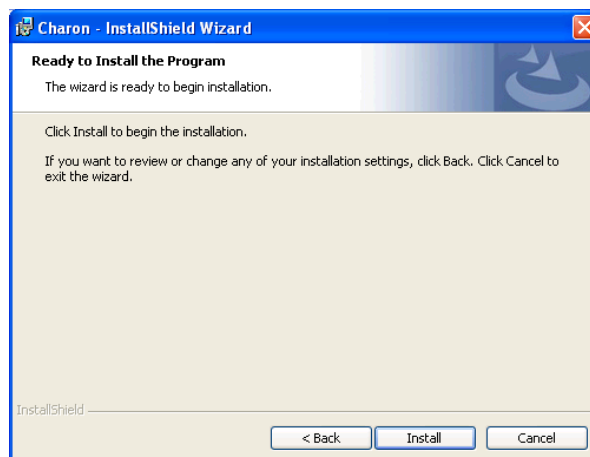


Figure 11: Ready-to-Install window

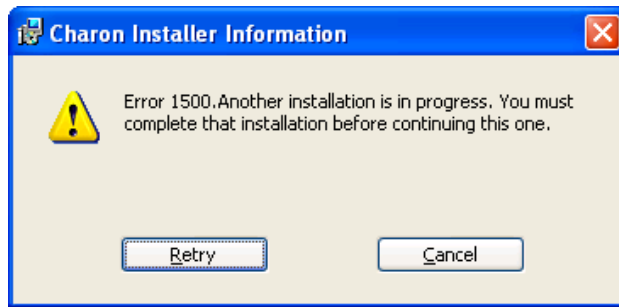


Figure 12: Error - another installation is in progress

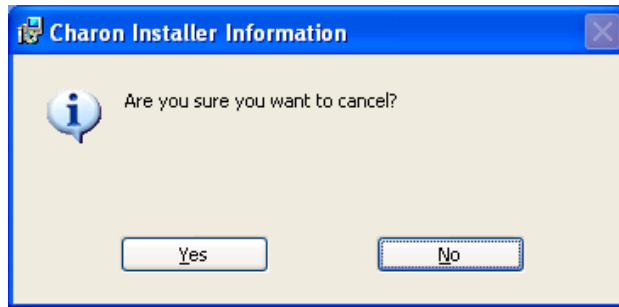


Figure 13: Confirmation of cancelling the installation

On Windows operating systems with UAC (User Account Control) the installation of Charon03 starts by opening the window shown in Figure 14. The first UAC message shown in Figure 15 is to be confirmed with YES, before Windows configures the installation of Charon03 (Figure 16). The icon of an UAC request appears as shown in the middle of Figure 17. Click on the blinking icon and confirm the next UAC message with YES (Figure 18).

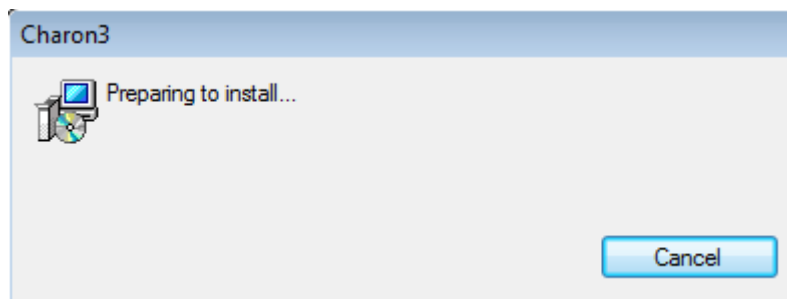


Figure 14: Prepare to install

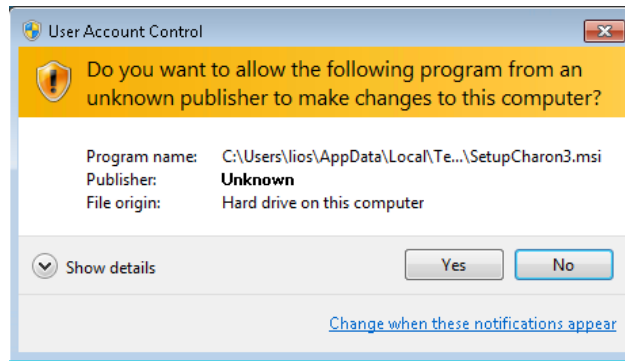


Figure 15: User Account Control (UAC) on Windows operating systems with UAC

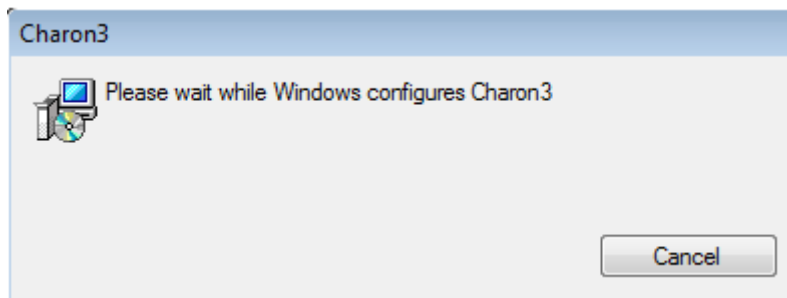


Figure 16: Windows 7 configuration of Charon03



Figure 17: UAC request on Windows operating systems with UAC

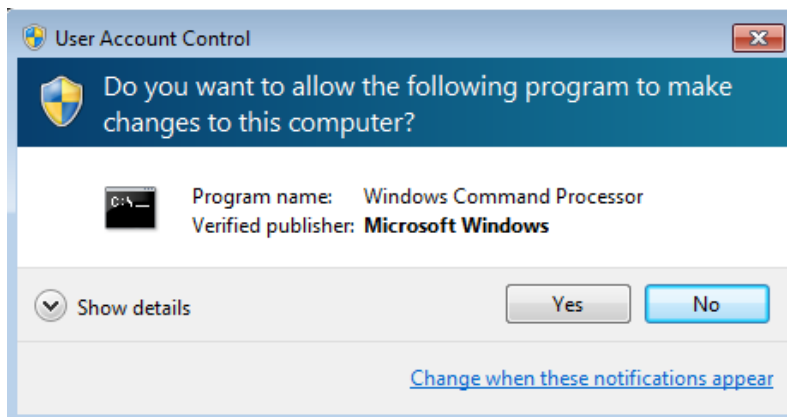


Figure 18: UAC Message on Windows operating system with UAC

The progress of the installation is indicated by a progress bar (Figure 19).

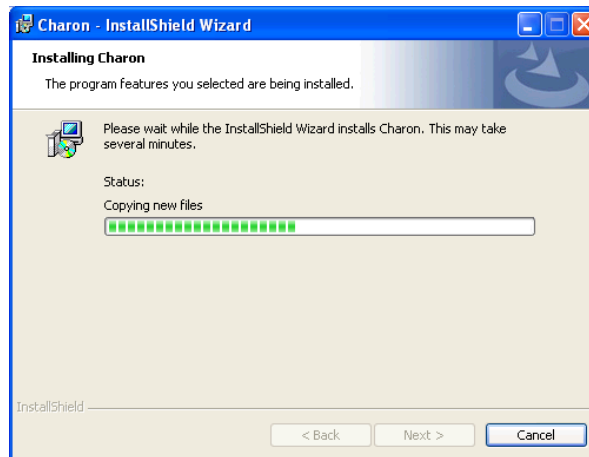


Figure 19: Progress bar during Charon3 installation

If the controller is not connected using USB, the ATTACH DEVICE hint (Figure 20) may appear. Press OK to continue.

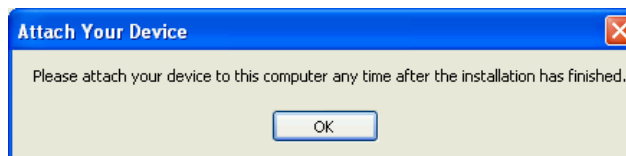


Figure 20: Attach device hint

When the installation is completed, finish the installation by pressing the FINISH button (Figure 21). The Start Menu of Windows then comprises a Charon3 folder including links to Charon3 and the License (Figure 22).

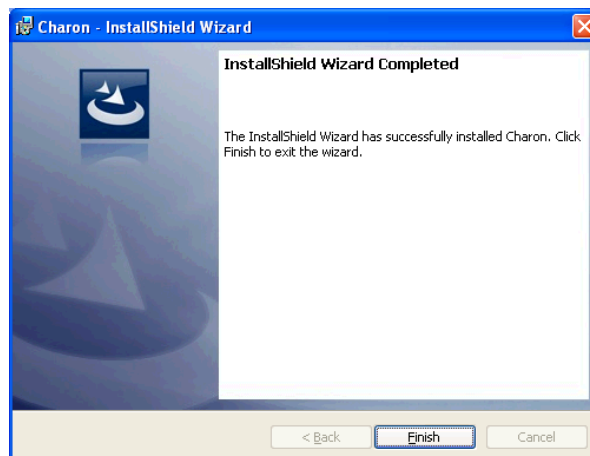


Figure 21: Finish installation window

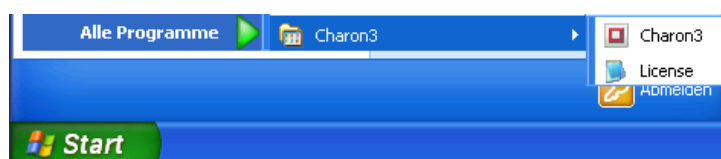


Figure 22: Charon3 entry in the START menu of Windows



3.3 Starting Charon3 and Setting of User Levels

Start Charon3 by clicking the Charon3 entry in the Windows START menu (Figure 22). If Charon3 is started the first time when the PC is operating with Win7 the Windows Firewall has to be configured as shown in Figure 23.

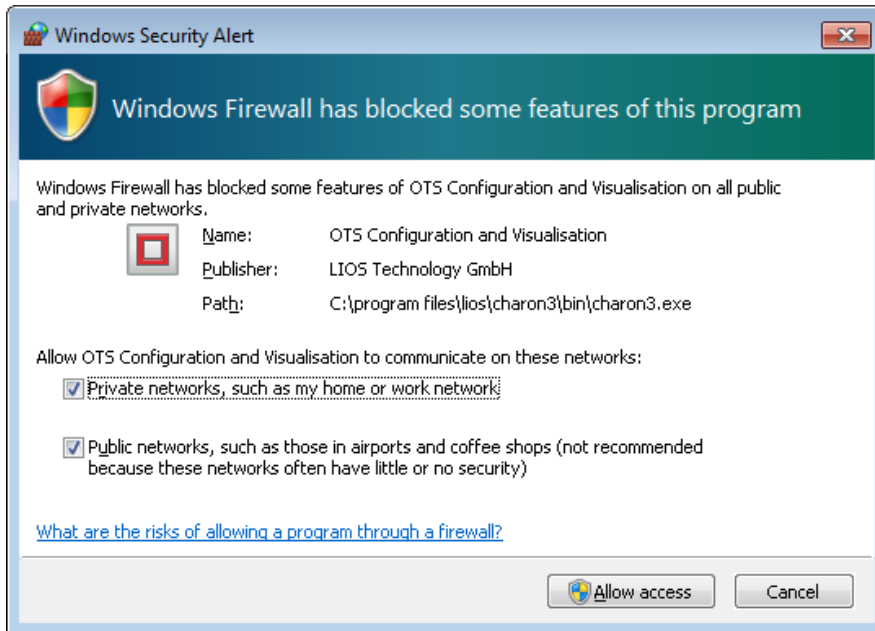


Figure 23: Initialization of the Windows Firewall

Initially the Charon3 window opens in the user level 1 mode (Figure 24).

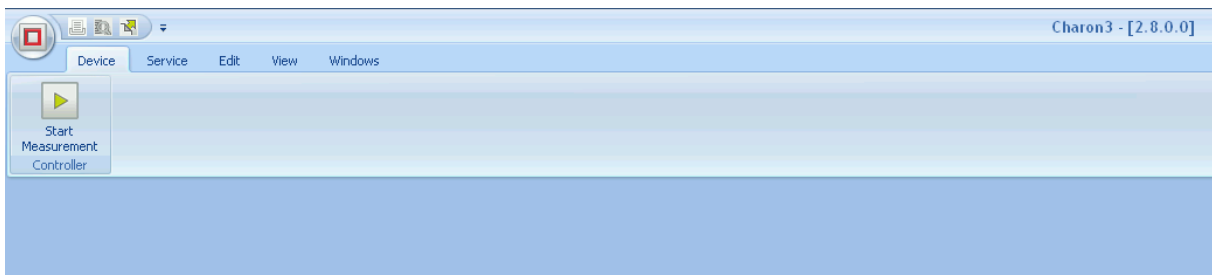


Figure 24: Charon3 window in the user level 1

Charon3 offers four user levels with different rights of the users (Table 3). When Charon3 is running, a different user level can be activated at any time by pressing Alt-Shift and the function keys given in Table 3. User name and password are required for the activation of levels 2 to 4. User names and passwords are provided to each customer in a separate document. The document is located in the subfolder LICENSES-PASSWORDS of the documentation binder shipped with the controller.

All further descriptions are based on user level 3 unless otherwise noted. Please consider the limitations of rights given in Table 3 when using other user levels.



Table 3: User levels

User Level	Activation	Rights
1	Alt-Shift-F1	Start measurement, data visualization
2	Alt-Shift-F2, User name, Password	Stop measurements
3	Alt-Shift-F3, User name, Password	Change controller settings
4	Alt-Shift-F4, User name, Password	Edit application settings

When user level 3 is activated, you can choose the SERVICE MODE (see section 7.3.2.1) in order to automatically start Charon3 in user level 3 without entering the password again.

Please change the display language from ENGLISH to your preferred language if you wish a display in another language (see section 0). The following instructions are based on English as display language.

The Charon3 window shows the DEVICE, SERVICE, EDIT, VIEW and WINDOWS tabs comprising a number of buttons for operating the controller and for visualization of temperature profiles as well as the TREE VIEW ().

The TREE VIEW shows a list of controllers previously connected to the PC. Controllers in the TREE VIEW are arranged by project. The controller entry comprises the controller name, the IP address if connected via Ethernet, the status and a description set by the user. The serial number is used as the default controller name. The STATUS column indicates if the laser is warming up, measuring or operating in the test mode. The time left for finalising the current measurement is indicated in a progress bar.

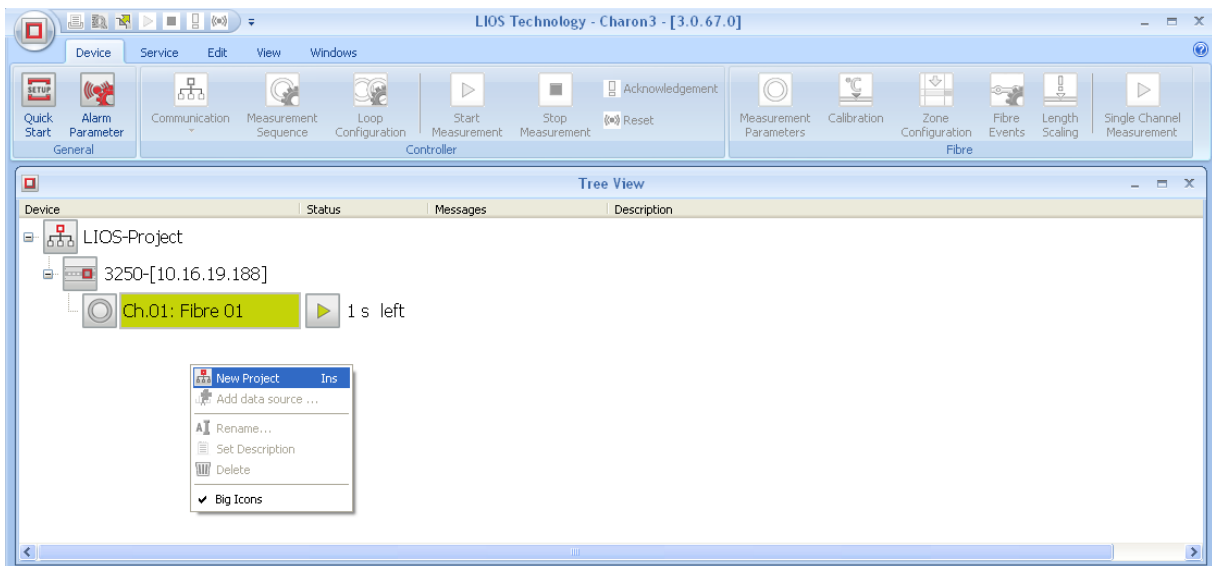


Figure 25: Charon3 window in user level 3

The nodes of the projects and controllers are visualized with a “+” and “-” sign. In the tree view they can be opened or retracted using the keyboard keys “+” and “-”, or by a mouse click. When a project node is opened the respective controllers are shown. When a controller node is opened the connected fibers are shown.



A right mouse click on a controller opens a pull down menu (Figure 26) offering a number of commands for operating the controller and for modifying its name, description and appearance in the TREE VIEW. Details on the commands are given in the sections describing the Charon3 tabs.

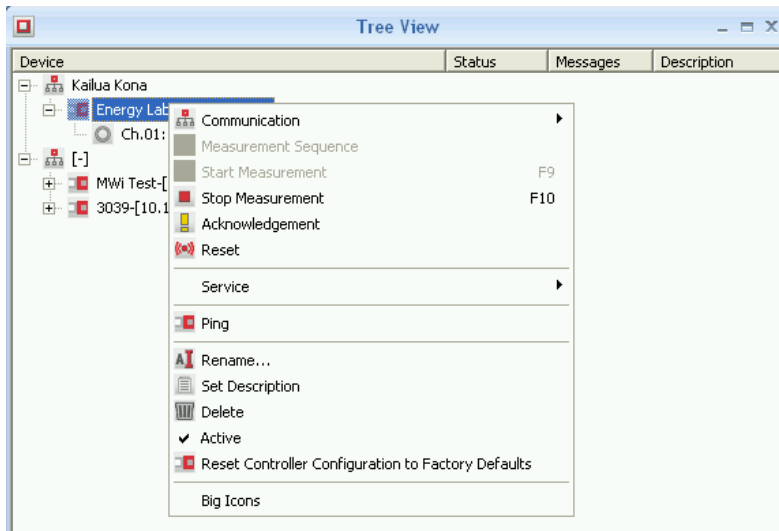


Figure 26: Tree view with pull down menu of a controller

The basic version of Charon3 can be installed on any suitable computer (see section 10.1). Without any additional license, it is only intended for commissioning and servicing of controllers. It automatically disconnects from all controllers after running for 4 hours. You can reconnect to the controllers after closing and restarting Charon3.

A visualization license for each controller is required for a permanent connection to Charon3.

3.4 Software Update

During installation, Charon3 automatically detects if there is another version of Charon3 already installed on the computer. If the installed version is lower than the one on your CD, you are asked to confirm the installation of the update or to remove Charon3 (Figure 27). Press NEXT to confirm your selection.

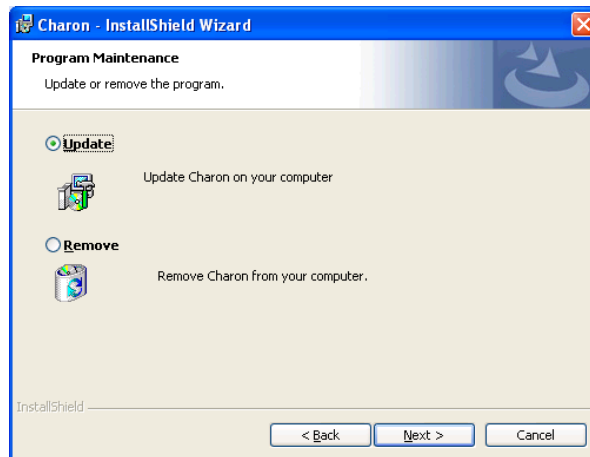


Figure 27: Confirmation of update

A downgrade to lower versions of Charon3 is not possible (Figure 28) because lower versions cannot read the database structure of higher versions. The only way to return to lower versions of Charon3 is a complete de-installation and re-installation of the lower version.

Caution!

All configuration and measurement data stored in the database will be lost by the downgrade. Make sure that you want to downgrade regardless the loss of data.

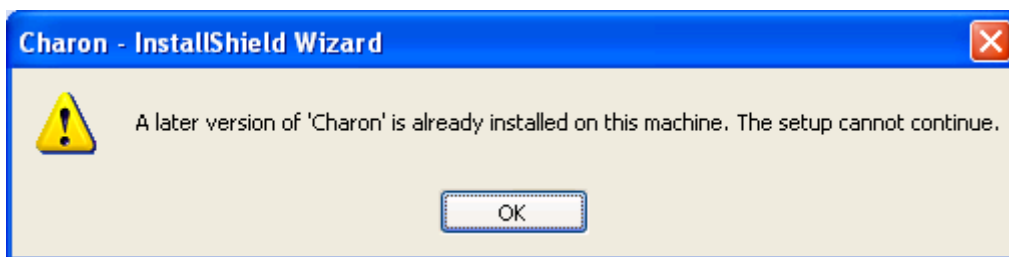


Figure 28: Information that a downgrade is not possible

3.5 USB Driver Installation

The USB driver is automatically installed during the installation of Charon3.

If you connect a controller to a PC without Charon3 which was never before connected to a controller, Windows automatically detects the controller indicated by the messages shown in Figure 29. Then, the Found New Hardware Wizard starts with the welcome message shown in Figure 30. Select NO, NOT THIS TIME and click NEXT to continue. Select INSTALL FROM A LIST OR SPECIFIC LOCATION in the next window of the wizard (Figure 31).

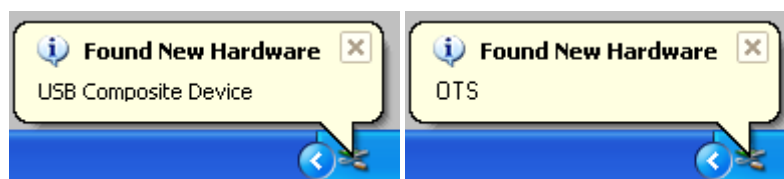


Figure 29: Window messages after connection to a controller using USB



Figure 30: Welcome message of the Found New Hardware Wizard

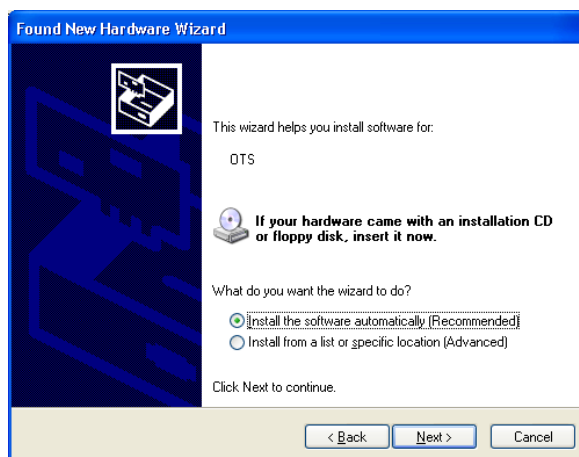


Figure 31: Selection of installation mode

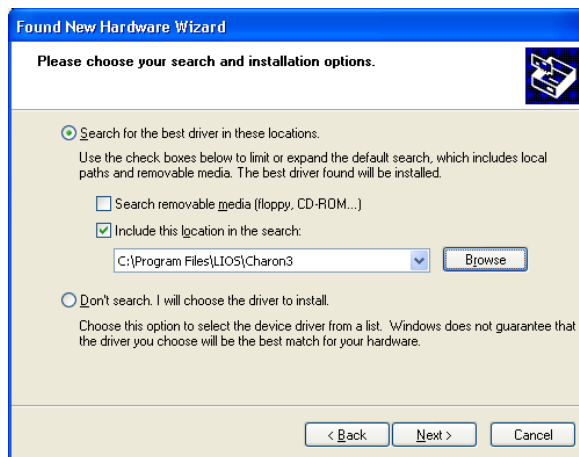


Figure 32: Selection of driver location



Select INCLUDE THIS LOCATION IN THE SEARCH in the next window of the wizard (Figure 32). Enter the path of the Charon3 installation on your computer and click NEXT. The further messages shown in Figure 33, Figure 34 and Figure 35 indicate successful completion of USB driver installation. Press FINISH to close the wizard.

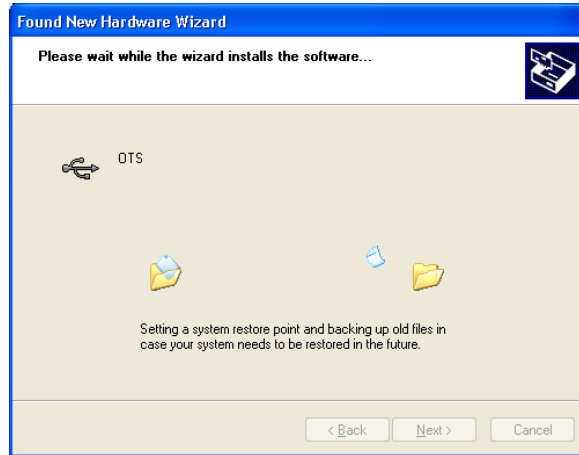


Figure 33: Hardware installation wizard.



Figure 34: Finish hardware installation wizard.

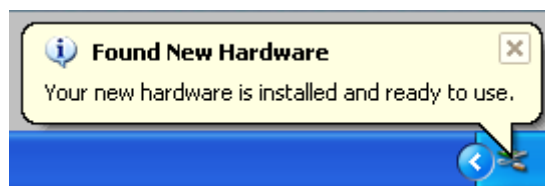


Figure 35: Hardware installed



4 Commissioning the Controller

4.1 Quick Start

QUICK START is a software feature for creating controller entries in the TREE VIEW of Charon3, for easy checking of proper controller hardware installation and for loading basic settings on the controller.

Install the controller and the sensor cable(s) as described in the Installation Manual. Turn on the controller.

Connect the controller and the computer with the Charon3 software installed directly using USB or cross-over Ethernet cables or connect both to a common local area network (LAN) using Ethernet patch cables.

Start Charon3, activate user level 3 (see section 3.3) and press the QUICK START button in the DEVICE tab of Charon3 (Figure 36) to start QUICK START.

The START window with the LIOS Technology logo in the centre (Figure 37) appears if no connected controller is detected. Please verify that the controller is switched on (POWER and SYSTEM READY LEDs on) and that the connection is established. The communication LED of the respective controller interface must indicate some data traffic after establishing a connection. If necessary, deactivate the Firewall or open the network ports required for the LIOS discover and communication protocols (Table 25). Use another interface if necessary.

The message SEARCHING FOR DEVICES will be automatically replaced by the CONTROLLER TABLE (Figure 38) if any controller(s) are detected. The detection of a new connection to the controller can take a few seconds.

When a controller has been newly connected via USB to a PC with running Charon3, QuickStart starts automatically.

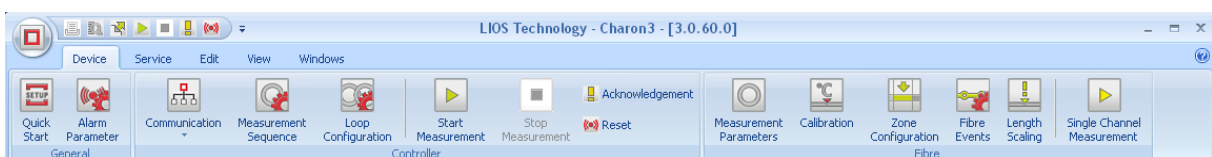


Figure 36: DEVICE tab with QUICK START button on the left



Figure 37: START window with message SEARCHING FOR DEVICES (being displayed if no controller is connected)



Figure 38: START window with CONTROLLER TABLE being displayed if controller(s) are connected

The CONTROLLER TABLE shows all connected controllers arranged by project. The following columns are displayed:

Table 4: Columns of the CONTROLLER TABLE and their content

Column	Content
Project	Name of the project the controller is used for. The default value is [-].
Name	Name of the controller. The default value is the serial number.
Model	Controller model as given on the nameplate.
S/N	Serial number of the controller.
Connection	Type of interface or IP address. Values are USB for USB interface or IP address if an Ethernet link is used. If the Ethernet interface is not yet configured the IP address 0.0.0.0 is displayed. In that case the setting of the connected Ethernet interfaces follows in the next Quickstart configuration dialogue, which appears automatically.



PROJECT and NAME of the controller can be changed after finishing the QUICKSTART in the TREE VIEW by right clicking with the mouse on the name, selecting RENAME in the pulldown menu and by entering the new value in the field.

A controller can be selected for checking the sensor cables by selecting the controller line in the table with the mouse. The check is started by pressing the NEXT button on the bottom of the window or by a double click on the controller entry.

If a controller is linked to a different Ethernet network than the computer with Charon3, the NETWORK SETTINGS – TCP/IP PROPERTIES window (Figure 39) opens. Otherwise, the CONTROLLER window (Figure 42) opens.

The network settings can be adjusted using the NETWORK SETTINGS menus. Alternatively, the controller can be connected using USB.

Caution!

Wrong network settings may lead to a loss of the connection to the controller and to other network failures. Please ask the local network administrator for support and the required information about the network settings.

Check the network settings of your computer before changing any network settings of the controller.

Press “WindowsKey-R” to open the execute prompt of windows. Enter “cmd.exe” to open the command line window of Windows. Enter the command “ipconfig /all” followed by the ENTER key. Make a note of IP addresses and subnet masks of the Ethernet adapter used for the connection, of the IP addresses of DNS server(s) and the DNS suffix.

Select the PRIMARY ETHERNET tab to configure a link over the Ethernet 1 socket or SECONDARY ETHERNET tab to configure a link over the Ethernet 2 socket. You can also configure both Ethernet interfaces if required. Enter the required information as given in Table 5. A warning sign (Figure 39, left) appears on the TCP/IP PROPERTIES tab as long as no network connection has been established.

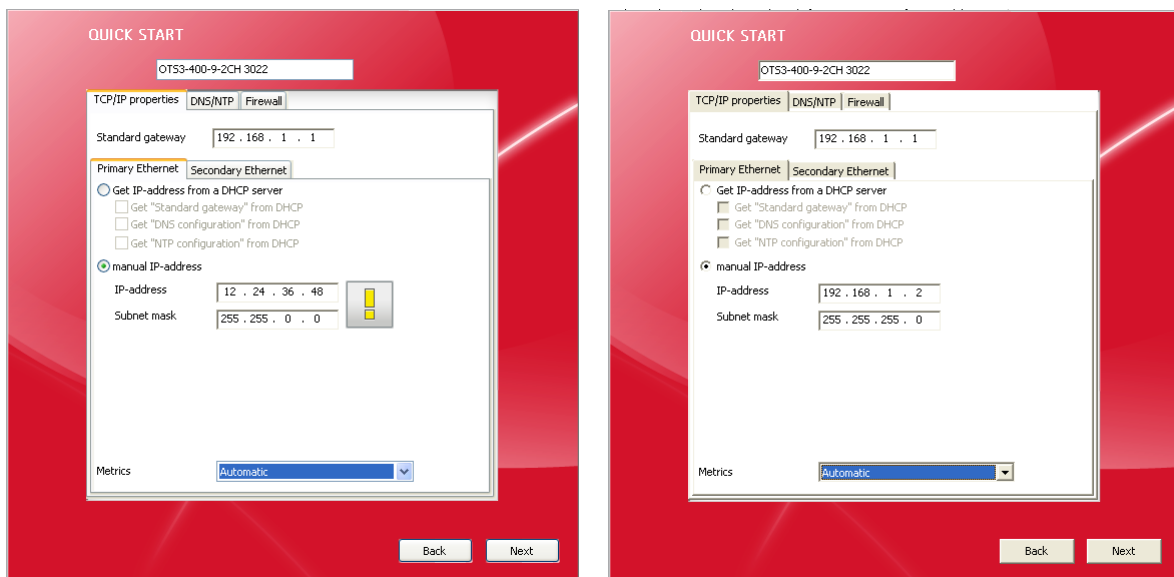


Figure 39: TCP/IP properties windows with (left) and without (right) warning sign



Table 5: TCP/IP properties settings

Parameter	Default	Custom
Standard gateway		Any change to be defined by the local network administrator.
IP address	From DHCP server	The IP address and subnet mask consists of 4 octets (numbers separated by dots). The controller must have the same subnet mask as your computer. Therefore, all octets where the corresponding new network mask octet is not zero must have the same value as your computer Ethernet adapter. At least one octet needs to be different from your computer and all other devices in the network. Ask the local network administrator to define an IP address.
Subnet mask	255.255.255.0	Same value as used Ethernet adapter of your computer.
Metrics	Automatic	Any change to be defined by the local network administrator.

Select the DNS/NTP tab (Figure 40) after finishing the TCP/IP properties settings. Enter the DNS server addresses detected at your computer. The DNS suffix of your computer can be entered in the DOMAIN field if requested by the local network administrator. A NTP server name or IP address defined by the local network administrator must be entered if automatic setting of the controller's time is required.

The controller clock is set to Universal Time (UT), a modern continuation of Greenwich Mean Time. Therefore, the time stamps of the controller can differ from local time. The UT time stamps do not depend on the controller location or seasonal local times. Depending on the operating temperature, the real time clock of the controller may show a slight shift over time. Synchronisation of time is essential to maintain the accuracy of time stamps over long periods. Synchronization with a NTP server is the most accurate method of synchronization. If the controller is not synchronized with a NTP server, it is synchronized with the clock of the commissioning PC when a new configuration is sent. UT time stamps are used in data transmitted to third party systems. Depending on the settings (see section 7.3.2.1), Charon3 displays and exports the data with local time stamps.

Select the FIREWALL tab (Figure 41) after finishing the DNS/NTP settings and enter the information provided by the local network administrator. A warning sign appears on the FIREWALL tab as long as no network connection has been established. Leave the NETWORK SETTINGS by pressing the NEXT button.

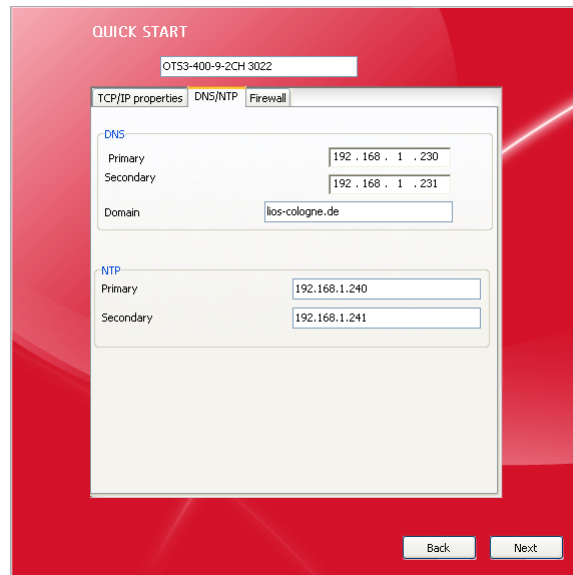


Figure 40: Network menu - DNS/NTP window

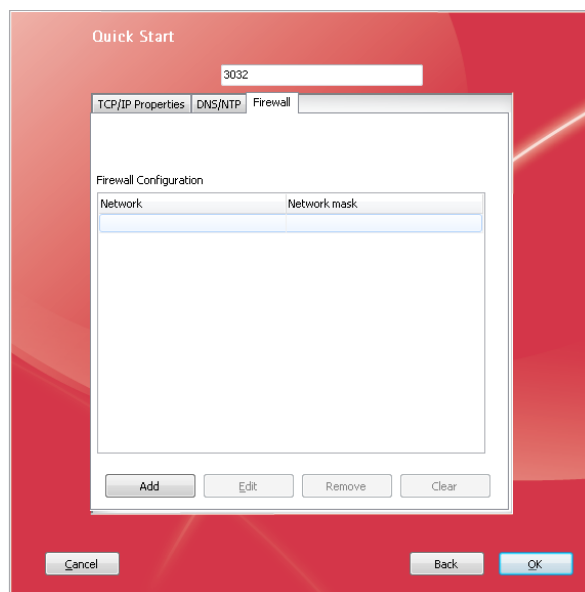


Figure 41: Network menu – Firewall window

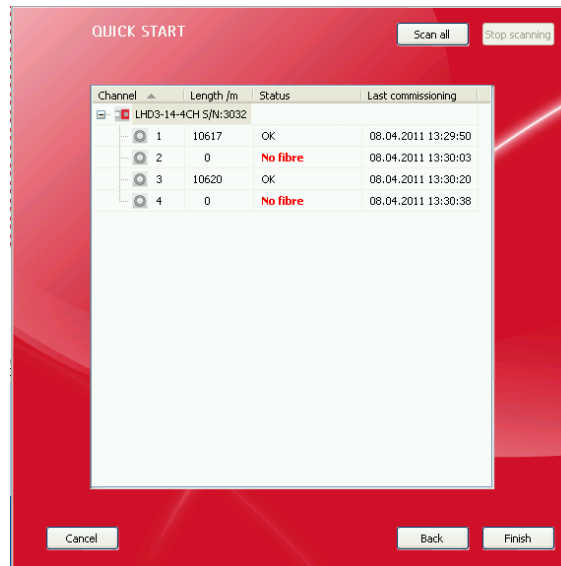


Figure 42: Controller window

The CONTROLLER window (Figure 42) shows the controller model and serial number on the upper left corner as well as a table of channels. The table of channels has four columns showing the channel numbers, the length and status of the connected fibre(s) as well as the date and time of the last commissioning. The controller checks length and status of each fibre during the scan. Before scanning, length and status are marked as “unknown”, during scanning they are marked as “scanning” and after scanning, length and status are displayed. The length is displayed in meters. A summary of status values is given in Table 6.

Table 6: Fibre status values and their interpretation

Status value	Interpretation
Unknown	Fibre is not yet checked.
Scanning	Fibre check is running.
OK	Fibre is ready for measurement.
No fibre / Fibre too long	No fibre was detected or the connected fibre is too long for the controller model.
Check connector	The fibre connector at the controller has too high attenuation or return loss. Please stop the measurement, clean and inspect the connector, plug it in again and re-scan the fibre. The connector needs to be replaced in case of severe damage of the ferrule endface (see Installation Manual).
Check termination	The fibre termination has too high return loss. Please stop the measurement and check the fibre termination (see Installation Manual).
Check fibre	The fibre may show high point losses and/or attenuation. Please use Charon3 to check the backscattering curves of the fibre (see section 5.4).

All channels or a single channel can be re-scanned after checking the connector or termination. Select the controller icon and click SCAN ALL to re-scan all fibres (Figure 43). Select a channel icon and click SCAN CHANNEL to re-scan a single channel (Figure 44). Scanning of channels can be terminated by selecting the controller icon and clicking STOP SCANNING.

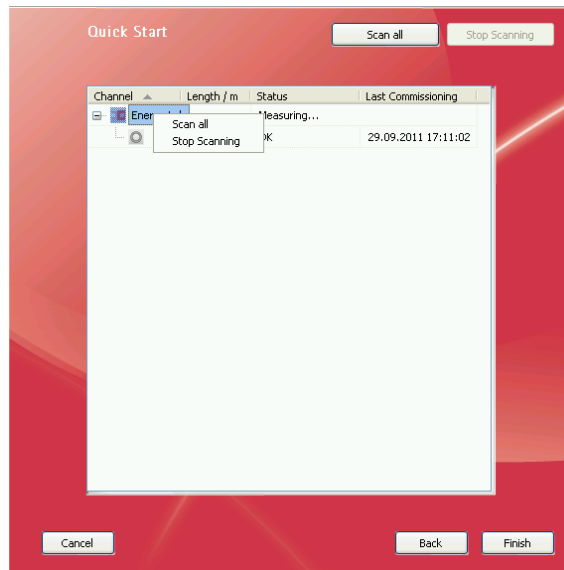


Figure 43: SCAN ALL and STOP SCANNING

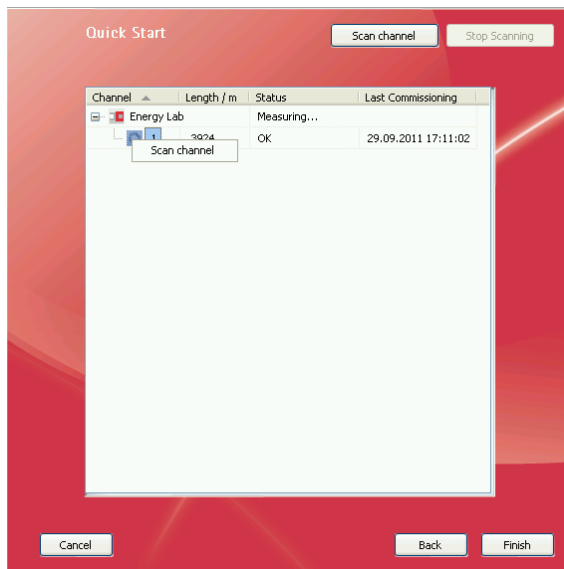


Figure 44: Scanning a single channel

Press the BACK button in the CONTROLLER window (Figure 42) if you want to check another controller. The controller settings are saved in the controller memory, and the CONTROLLER TABLE window (Figure 38) is displayed again. No new entry of the already checked controller is added to the tree view in this case.

In the lower right corner of the CONTROLLER window (Figure 42) appears a NEXT button if no zones were previously defined on the controller. A FINISH button appears if the are zones defined.

Press the NEXT button to open the APPLICATIONS window (Figure 45). A list of pre-defined applications is shown including their default zones. Each application may define a feed zone and a runout zone as well as a number of alarm zones. The number of alarm zones is automatically set according to the length of the fibres minus the lengths of the feed and runout zones. Select an application to define the zones on the fibres of the controller and to use the other settings of the application setting. Application settings are described in detail in section 7.3.2.1.

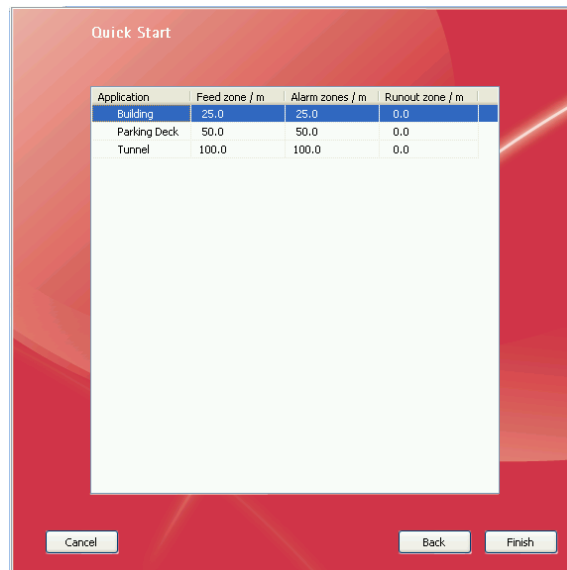


Figure 45: Applications window

Quit QUICK START by pressing the FINISH button on the lower right corner of the CONTROLLER window (Figure 42). Controller settings are saved in the controller memory and will be available for Charon3 software. A report on the QUICK START results and controller settings is saved as PDF document in your Charon3 default report folder: C:\OTSDData\Reports. The folder path may be changed in the application settings.

After finishing the QUICKSTART the controller is added to the tree view as shown in Figure 46.

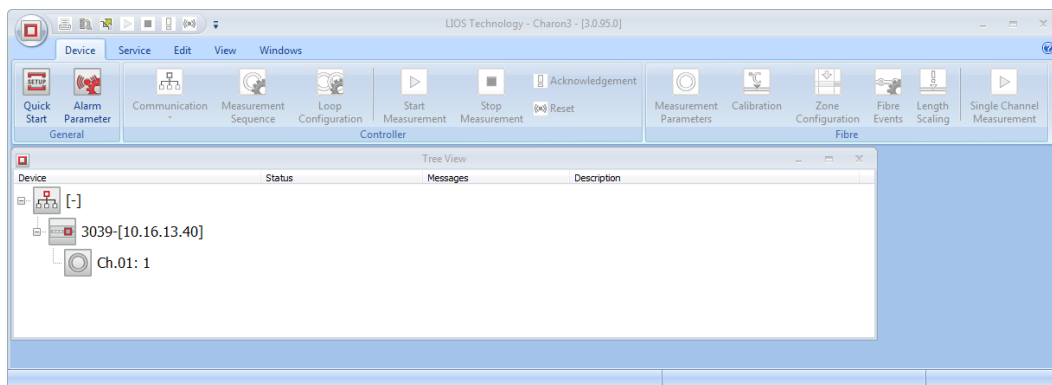


Figure 46: Tree View after finishing the QUICKSTART

4.2 Trigger Parameters

Trigger parameter sets of two different types, the alarm parameter sets and the control parameter sets, can be defined using the Charon3 button ALARM PARAMETER in the DEVICE tab. The trigger parameter sets are stored in the Charon3 database.

4.2.1 Alarm Parameter Sets

The tab ALARM PARAMETER SET (Figure 47) shows a list of previously defined alarm parameter sets. A number of alarm parameter sets are shipped with Charon3. Further sets can be defined by the operator. Several parameter sets have been approved by certified bodies according to national standards. The approved alarm parameter sets are write-protected.



Warning!

Only approved parameter sets must be used for operating the controller in compliance with the respective standards. Any use of other alarm parameter sets is not covered by the approvals!

Table 7: Alarm parameter sets

Alarm parameter set	Standard	Country / Region	Response class / Temperature Rating	Applications	Maximum application / maximum installation temperature	Approved by
EN54-22 A1N	EN54-22	Europe	A1N	Room protection, e.g. road tunnels, buildings	50°C	VdS
EN54-22 BN	EN54-22	Europe	BN	Local protection, e.g. equipment operated up to 65°C	65°C	VdS
EN54-22 CN	EN54-22	Europe	CN	Local protection, e.g. equipment operated up to 80°C	80°C	VdS
UL521 Fixed 135F	UL521 / ULCS 530	USA / Canada	Ordinary 135°F-174 °F (57.2°C- 78.9°C)	Heat detector also suitable as heat actuated device for special application	100°F (37.8°C)	UL
UL521 Fixed 160F	UL521 / ULCS 530	USA / Canada	Ordinary	Heat detector also suitable as heat actuated device for special application	100°F (37.8°C)	UL
UL521 Fixed 190F	UL521 / ULCS 530	USA / Canada	Intermediate	Heat detector also suitable as heat actuated device for special application	150°F (65.6°C)	UL
UL521 ROR 15F / 20F	UL521 / ULCS 530	USA / Canada	Intermediate	Heat detector also suitable as heat actuated device for special application	150°F (65.6°C)	UL

Alarm parameter sets can be created or modified in the ALARM PARAMETER SET folder.

For unambiguous identification, a unique name should be defined for any new alarm parameter set.

We do not recommend modifying or deleting any parameter set previously used in a controller setting. If a used parameter set must be changed, it is automatically uploaded to any connected control-



ler. The original set is further used in the unconnected controllers. When Charon3 reconnects to a controller using obsolete parameter sets, you must select the option to upload the configuration data from Charon3 to the controller. Any download of an obsolete parameter set from a controller will create a double entry in the parameter set list with the same name for different settings!

The definition of alarm parameter sets starts with the unique NAME followed by a DESCRIPTION (Figure 47).

The maximum temperature alarm criterion is specified by the alerting level T_{Max} (Figure 2). Temperature readings exceeding T_{Max} cause an alarm.



The minimum temperature alarm criterion is specified by the alerting level T_{Min} . Temperature readings going below T_{Min} cause an alarm. The minimum temperature alarm is not used for fire detection, but can be used for detecting other events such as icing.

In user level 3, the entry MIN-MAX CYCLES is shown specifying the number of measurement cycles used for internal processing of the maximum and minimum temperature alarm criteria. The default setting is 10. The number can be increased up to 20 to further reduce the temperature noise, or it can be reduced down to 1 to shorten the response time. If the entry MIN-MAX CYCLES is set to 1, the alarm is immediately triggered when a single temperature value fulfils the trigger level. Higher values of the entry MIN-MAX CYCLES reduce the probability of false alarms without significantly delaying the alarm response. The default setting of 10 is well suited for most applications. We recommend not changing this value.

Up to three rate-of-rise criteria (Figure 4) can be used in an alarm parameter set. They are specified by the pairs of entries $\Delta T_x/^\circ C$ and $\Delta t_x/s$. $\Delta T_x/^\circ C$ specifies the minimum temperature increase over the time period $\Delta t_x/s$ triggering an alarm. The time period for evaluating the rate-of-rise ($\Delta t_x/s$) should be larger than 5 times the cycle time and smaller than 20 times the cycle time.

Two further alarm criteria can trigger alarms in case of high deviations of local temperatures from the average temperature of a zone. These criteria are called HOT SPOT ΔT and COLD SPOT ΔT . The parameter HOT SPOT ΔT specifies the minimum temperature difference between the hot spot and the zone average triggering an alarm (Figure 3). The parameter COLD SPOT ΔT specifies the minimum temperature difference between the zone average and the cold spot triggering an alarm. Both criteria are not used in EN 54-22 compliant applications.

The ALARM DELAY specifies the number of measurement cycles used for the validation of an alarm. An ALARM is released immediately after the first triggering event if the ALARM DELAY is set to zero. An ALARM DELAY of 1 means that an alarm is only released if it is triggered in two subsequent measurement cycles.

Figure 47 shows the EN 54-22 and UL521 compliant and additional alarm parameter sets. Default parameter sets shipped with the controller are read-only and marked with the alarm icon . Custom parameter sets can be edited or deleted and are marked with the edit icon .



Name	Description	Tmax / °C	Tmin / °C	ΔT1 / °C	Δt1/s	ΔT2 / °C	Δt2/s	ΔT3 / °C	Δt3/s	Hot Spot Δ Tj / °C	Cold Spot Δ Tj / °C	Alarm Delay
EN 54-22 A1N	EN alarm parameter set for buildings	60,00	---	20,80	96,00	24,00	160,00	24,00	320,00	---	---	1
EN 54-22 BN	EN alarm parameter set for equipment up to 65°C	77,00	---	---	---	---	---	---	---	---	---	1
EN 54-22 CN	EN alarm parameter set for equipment up to 80°C	92,00	---	---	---	---	---	---	---	---	---	1
Fast	Fast measurement	50,00	---	6,00	41,00	9,00	105,00	12,00	153,00	10,00	---	1
Pre-Alarm	Pre-alarm parameter set	45,00	---	6,00	57,00	8,00	105,00	10,00	153,00	---	---	0
Test Mode	Test mode parameter set	40,00	---	5,00	57,00	8,00	105,00	10,00	153,00	8,00	---	0
Tunnel Portal	Tunnel portal parameter set	60,00	---	11,00	57,00	12,00	105,00	14,00	153,00	20,00	---	1
Tunnel Standard	Tunnel standard parameter set	55,00	---	7,00	57,00	9,00	105,00	12,00	153,00	15,00	---	1
ULS21 Fixed 135F	Fixed maximum temperature 135F	57,22	---	---	---	---	---	---	---	---	---	0
ULS21 Fixed 160F	Fixed maximum temperature 160F	71,11	---	---	---	---	---	---	---	---	---	0
ULS21 Fixed 190F	Fixed maximum temperature 190F	87,78	---	---	---	---	---	---	---	---	---	0
ULS21 ROR 15F / 20F	Rate of rise 15F / 20F	---	---	9,00	315,00	---	---	---	---	---	---	1
Ultrafast	Ultra Fast measurement	50,00	---	5,00	29,00	9,00	53,00	12,00	81,00	10,00	---	1

Figure 47: Alarm parameter set tab of the alarm / control parameter sets window

4.2.2 Control Parameter Sets

The folder CONTROL PARAMETER SET shows a list of previously defined control parameter sets. Control parameter sets can be created or modified using the Charon3 software.

Control parameter sets can be used to specify a two-level control for activation and deactivation of a self-reset output.

A unique name should be defined for each control parameter set considering the restrictions already mentioned for the alarm parameter sets.

Name	Description	Activation Temperature	Deactivation Temperature / °C
Ice Warning	Ice warning	3,00	6,00
Ventilation	Ventilation	30,00	25,00

Figure 48: Control parameter set tab

Control parameter sets were not evaluated by UL / ULC or Vds.

4.3 Communication Settings

Communication settings (event configuration, network settings, LON protocol, other protocols, locked mode) are accessible via the COMMUNICATION button in the DEVICE tab (Figure 49).

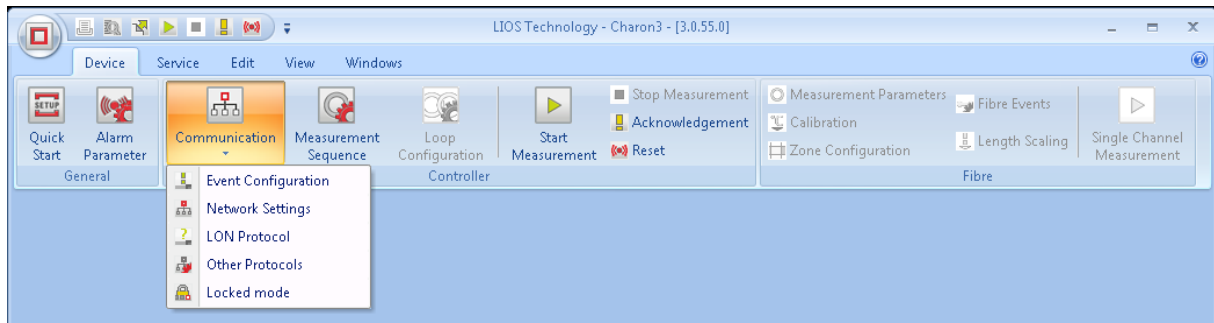


Figure 49: Communication settings pull down menu

4.3.1 Event Configuration

Events are occurrences of alarms, faults, warnings or specific user actions. By clicking EVENT CONFIGURATION in the COMMUNICATION button menu, the EVENT CONFIGURATION window opens where the handling of these events can be defined.

At the GENERAL tab (Figure 50), you can modify settings for the buzzer, the power up mode, the pre-alarms and the fibre beak behaviour. The buzzer can be activated for the ping connection test only or additionally for alarms, faults and pre-alarms.

You can choose to CONTINUE MEASUREMENT or to STOP MEASUREMENT in case of a fibre break.

Caution!

For fire safety applications, continuation of measurements in case of a fibre break is required by the standard EN 54-22 and UL521 / ULC S530. Therefore, the option CONTINUE MEASUREMENT must be chosen to operate the controller in compliance with these standards.

UL /ULC evaluated the standard LHD3-series controllers only for ordinary locations, not for the use in explosive atmosphere. The VdS approval covers the standard LHD3- and LHD3S-series controllers as well as the models with EX-option (EXT3 and EXT4).

For applications in explosive areas, the operator has to proof if the EX rating of the controllers with EX-option is appropriate for permitted use. We recommend choosing STOP MEASUREMENT as fibre break behaviour. For fire safety applications in explosive areas, the user has to choose the proper fibre break behaviour option based on a risk analysis of his application.

You can choose three different modes for operation after power up: NO MEASUREMENT, CONTINUATION OF MEASUREMENTS STARTED BEFORE POWER OFF or START MEASUREMENTS AUTOMATICALLY. We recommend using the continuation or automatic start of measurements for fire detection applications to comply with VdS and UL / ULC.

You can choose using SELF-RESET or MANUAL RESET outputs for pre-alarms. If you choose self-reset outputs, the outputs are reset automatically when the triggering condition is no longer present. Otherwise, the outputs stay activated until a reset is carried out. This is the same behaviour as for alarms.

**Caution!**

A change of the mode of the pre-alarm outputs causes a deletion of pre-alarm outputs which already have been assigned to an event.. Therefore it is strictly recommended to configure the pre-alarm output mode before any pre-alarm output is configured for use.

If you check the box DEACTIVATE SELF-RESET OUTPUTS ON RESET, the self-reset outputs are not only deactivated when the triggering condition is no longer present, but also by a reset.

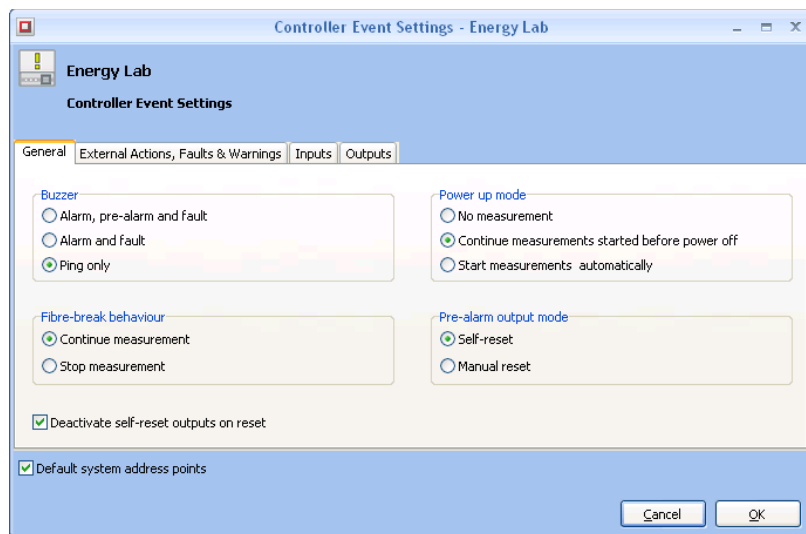


Figure 50: Event configuration, general tab

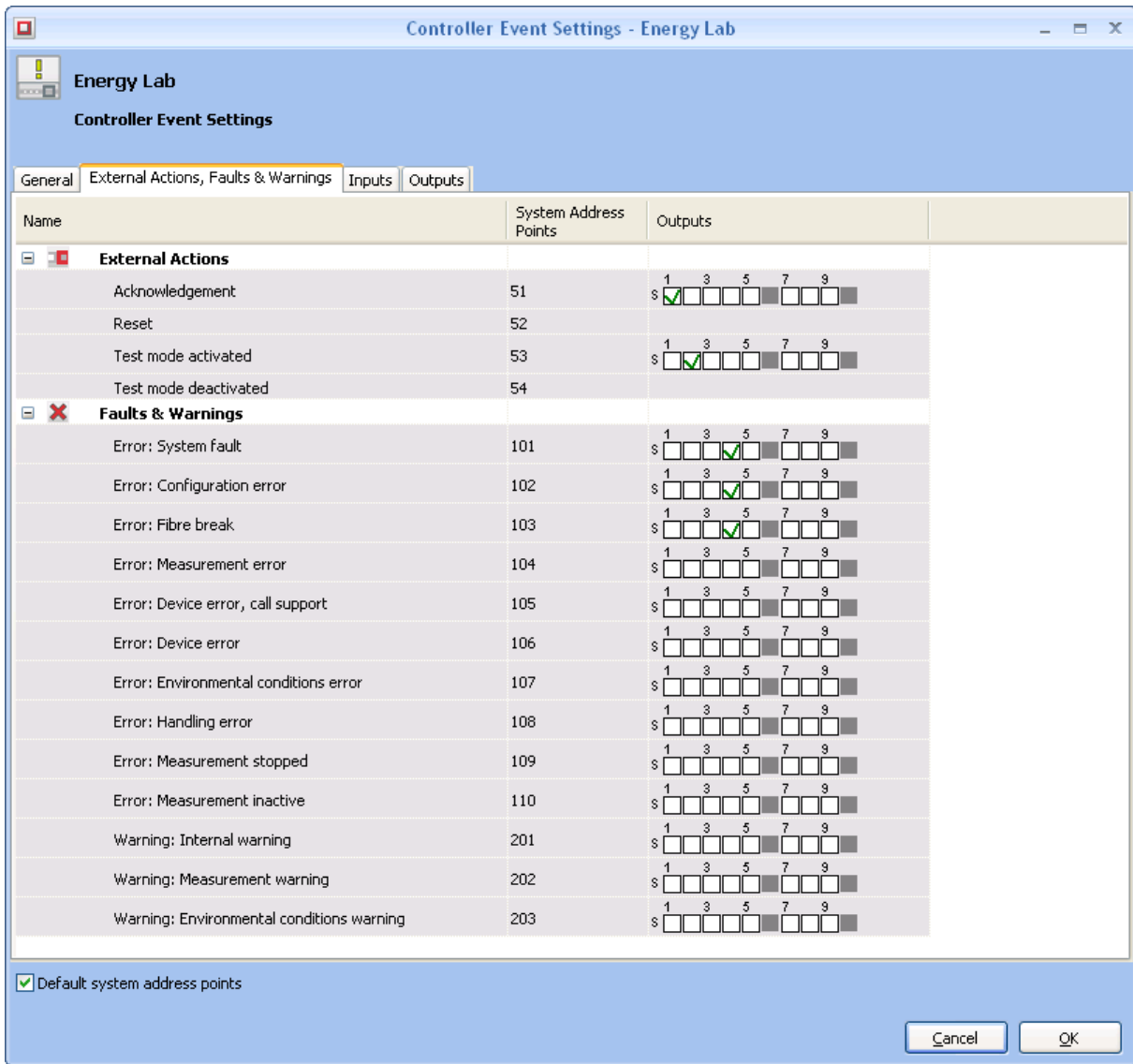


Figure 51: External Actions, Faults & Warnings tab

Within the EXTERNAL ACTIONS, FAULTS & WARNINGS tab (Figure 51) of the EVENT CONFIGURATION window, system address points as well as outputs can be assigned to specific events and external actions. An output is selected by clicking on the respective checkbox. Certain outputs may not be available for (manual reset) faults and warnings since they are already used for (self-reset) controls. These outputs are presented in grey and cannot be checked.

System address points have a default assignment as given in Table 22, Table 23 and Table 24. The assignment can only be changed after un-checking the box at the left lower corner of the EXTERNAL ACTIONS, FAULTS & WARNINGS tab (Figure 51).

Caution!

We strongly recommend not changing the system address point assignments in order to maintain consistent settings for all controllers.



For changing system address points, double click the cell in the SYSTEM ADDRESS POINT column of the event of interest. Enter an address point number between 0 and 255. Address points outside this range and address points already assigned to other events are not accepted by Charon3.

You may return to the default system address points by rechecking the box at the lower left corner. Then, a message indicating that all address points are going to be set to default values has to be confirmed (Figure 52).

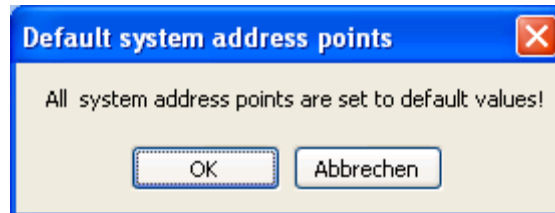


Figure 52: Message: System address points are set to default values

If an event occurs during controller operation, the specified address point is sent via the interfaces and the assigned outputs are activated.

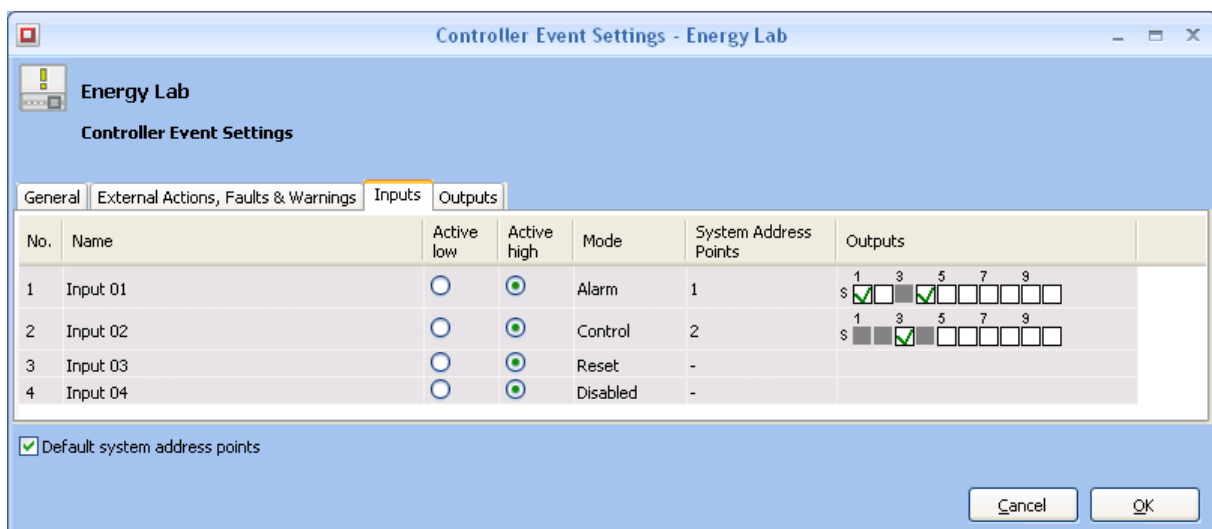


Figure 53: Inputs tab

At the INPUTS tab (Figure 53) of the EVENT CONFIGURATION window the active state and mode of each input can be set. Furthermore, system address points and outputs can be assigned to the inputs.

The active state can be selected using the columns ACTIVE LOW and ACTIVE HIGH. Active high means that the input is activated when a voltage in the specified range is applied, and that the input is deactivated when no voltage is applied. Active low has the opposite assignments.

In the SYSTEM ADDRESS POINT and the OUTPUT columns, system address points and outputs can be assigned to the inputs. To edit an address point, uncheck the DEFAULT SYSTEM ADDRESS POINTS box, double click the respective cell and enter the new value. Restrictions given above for other system address points also apply to the system address points of inputs. To assign outputs, check the corresponding box by left clicking on it.



Eight different modes can be selected in the INPUT MODE column:

- **ACKNOWLEDGE**
An activation of the input triggers an acknowledgement. The buzzer is switched off and an address point is sent.
- **ACTIVATE TEST MODE**
The controller is operated in the test mode as long as the input is active.
- **ALARM**
An activation of the input triggers an alarm. The alarm remains active after deactivation of the input until the controller is reset.
- **BUZZER/LED TEST**
The buzzer and all LEDs are switched on for a few seconds after activation of this input.
- **CONTROL**
The input is used for control purposes. Assigned outputs are self-reset and hence deactivate if the input is deactivated.
- **DEACTIVATE BUZZER**
The buzzer is switched off and – regardless the configuration - does not react on alarms, pre-alarms or system faults as long as the input is active.
- **DISABLED**
The input is not utilized.
- **RESET**
An activation of the input triggers a reset.

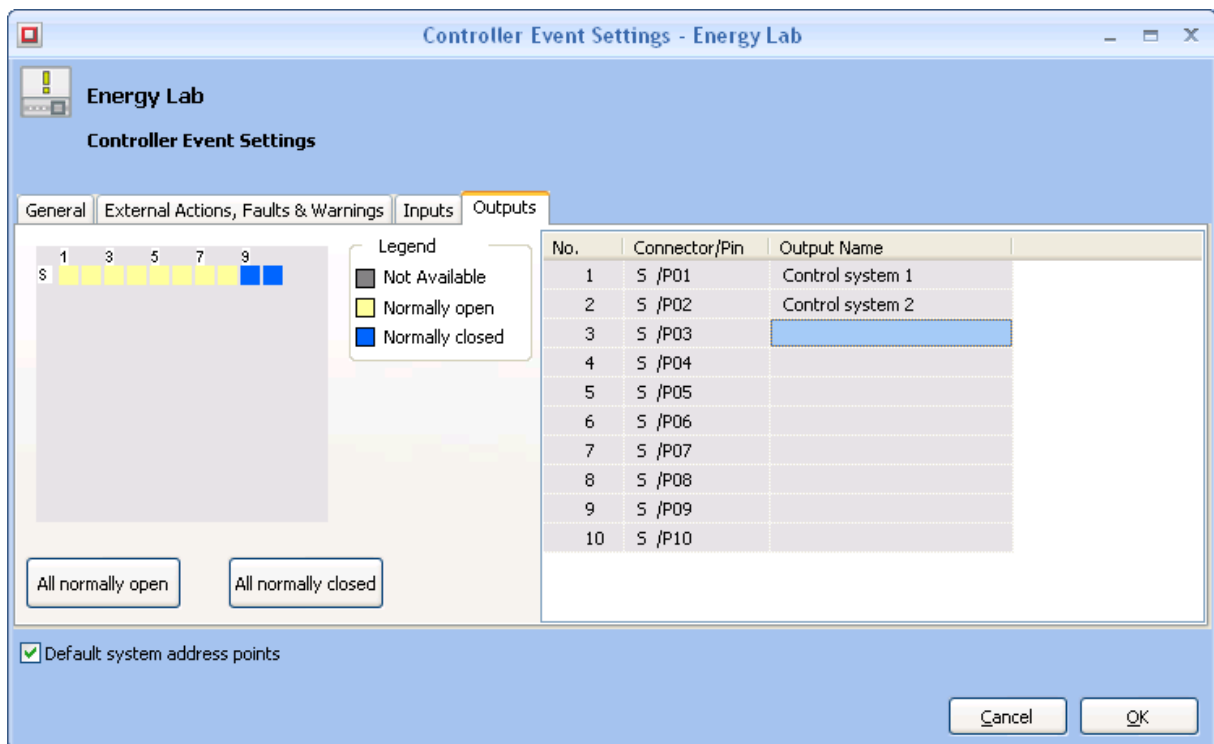


Figure 54: Outputs tab

At the OUTPUTS tab (Figure 54) of the EVENT CONFIGURATION window, the active state of each output and the output name can be set. The setting of each output is represented by the colours in the



diagram on the left side. Light yellow indicates a normally open state and blue a normally closed state. Clicking of a box changes the active state of the respective output. By clicking the ALL NORMALLY CLOSED or ALL NORMALLY OPEN buttons, all outputs are set to the normally closed or normally open operation, respectively. Output names can be entered into the field on the right side.

4.3.2 Network Settings

TCP/IP properties, DNS/NTP and firewall settings can be changed using the NETWORK SETTINGS window available via the COMMUNICATION button in the DEVICE tab.

Details on the network settings are given in chapter 4.1 QUICK START.

4.3.3 LON Protocol Settings

The controller transmits messages and data using the LON protocol defined in the Communication manual, Part I. LON protocol settings can be changed using the LON PROTOCOL window (Figure 55) available via the COMMUNICATION button of the DEVICE tab.

The LON protocol of the LHD3 controller offers additional features compared to previous LHD systems. The LHD3 controller can be operated using the LON protocol versions shown in Table 8 in order to simplify the replacement of previous systems by a LHD3 controller. However, restrictions regarding number of channels, distance range, zones, pre-alarms, control outputs and address points apply due to the limited capabilities of the earlier versions of the LON protocol. Therefore, use the default OTS3 protocol in all new projects.

After selection of another protocol version than OTS3, a warning window (Figure 56) opens showing the restrictions of the other protocol versions and information on the deletion of all zones. You need to confirm the information in order to activate the other protocol version.

Table 8: LON protocol versions

Protocol version	OTS3 (default)	5505	2005	1056
Channels	all options	1, 2 or 4	1	1
Distance range	all models	1, 2 or 4 km	1, 2 or 4 km	1, 2 or 4 km
Zones	1000 / channel	128 / channel	128	128
Pre-alarms	yes	no	no	no
Control outputs	yes	no	no	no
Fibre address points²	3100 / channel	256 / channel	256 fibre and system address points	256 fibre and system address points
System address points¹	256	256	not available	not available
Default system address points	available	not available	not available	not available
CRC	yes	yes	yes	no
Compatible controller models	OTS3/LHD3	OTS-X	OTS-X (1 channel) OTS-100	OTS-100
Compatible firmware versions		55xx	1065, 2005, 4000 (PC2005)	1056

² The table shows the capabilities of the protocol versions. Lower numbers of address points may be used depending on the software and the hardware options (e.g. inputs and outputs).



You can also select whether the LON messages are transmitted via Ethernet (TCP) and/or RS232 interfaces, which TCP port is used, and the baud rate for RS232 transmission (Figure 55). You can also assign a unique LON ID for identification of the LON messages of the controller.

Earlier LON protocol versions can be used to transmit data to Charon_02³ after activating the checkbox below the LON ID. In Charon_02, a controller with the same LON ID and zones needs to be configured in order to read the data. This enables common visualization of data from OTS3 and OTS-X using Charon_02.

In the lower part of the window, you can select which data shall be transmitted via LON. Zone data are only transmitted if zones are defined on the fibres connected to the controller and respective boxes in the window are checked. The transmission of temperature and backscattering profiles requires an optional license.

Press OK to confirm the setting or CANCEL to exit without saving the changes.

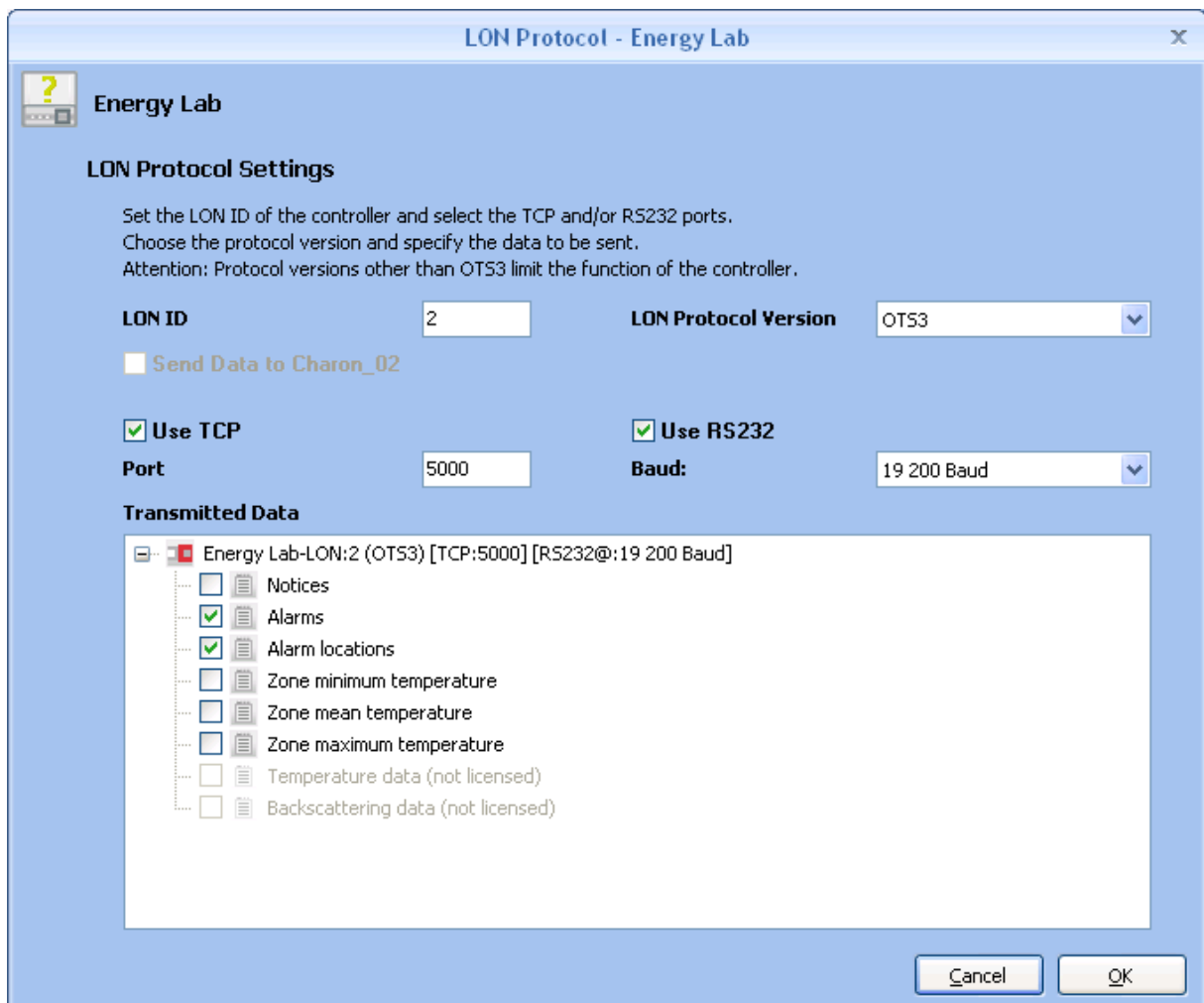


Figure 55: Communication settings: LON protocol

³ Charon_02 is a previous version of Charon3 that was shipped with controllers of the OTS-X series.

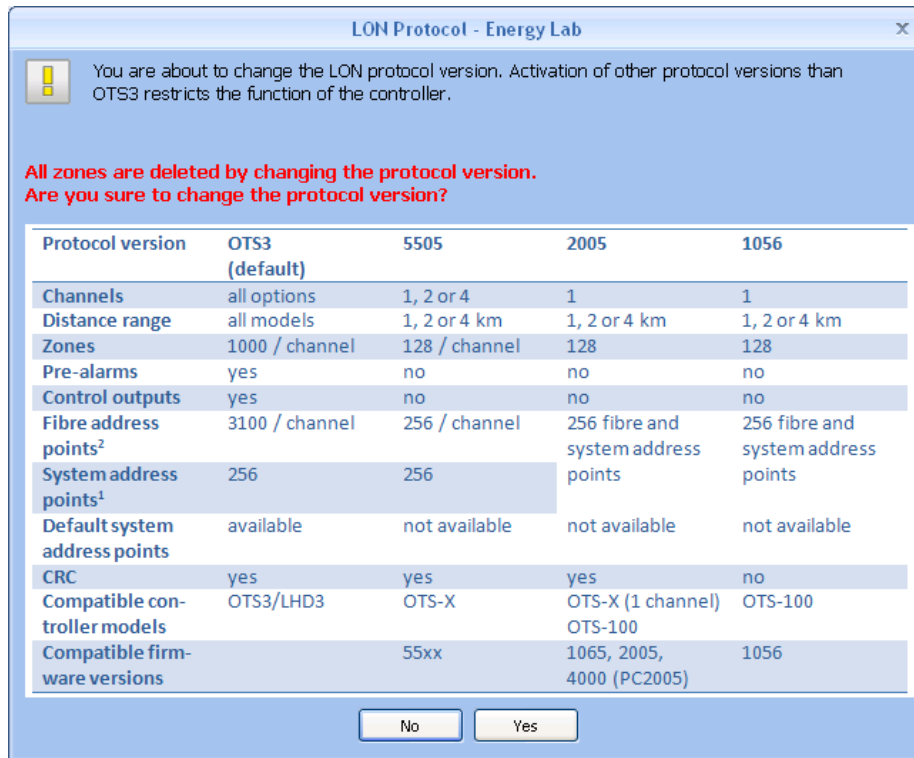


Figure 56: LON protocol change warning

4.3.4 Other Protocols Settings

The XML protocol is available on any controller. Further protocols can be available in dependence on installed licenses. The activation and settings of XML and the further protocols can be changed using the OTHER PROTOCOLS window (Figure 57) available via the COMMUNICATION menu in the DEVICE tab (Figure 49).

For instance, the configuration of the XML protocol (Figure 58) allows choosing:

- the TCP port to be used,
- the use of the TLS encryption
- the use and mode of data buffering by the CF memory card,
- the selection of data to be transmitted (depending on licenses), and
- the firewall settings.

Details on the individual protocols are given in the respective parts of the Communication Manual. In the configuration menu of the dedicated protocol a filter mask can be chosen to select the data to be transmitted. Selected data is transmitted as soon the data is available. This may require additional configuration steps. For example, zone average values are available only, if a zone configuration has been programmed.

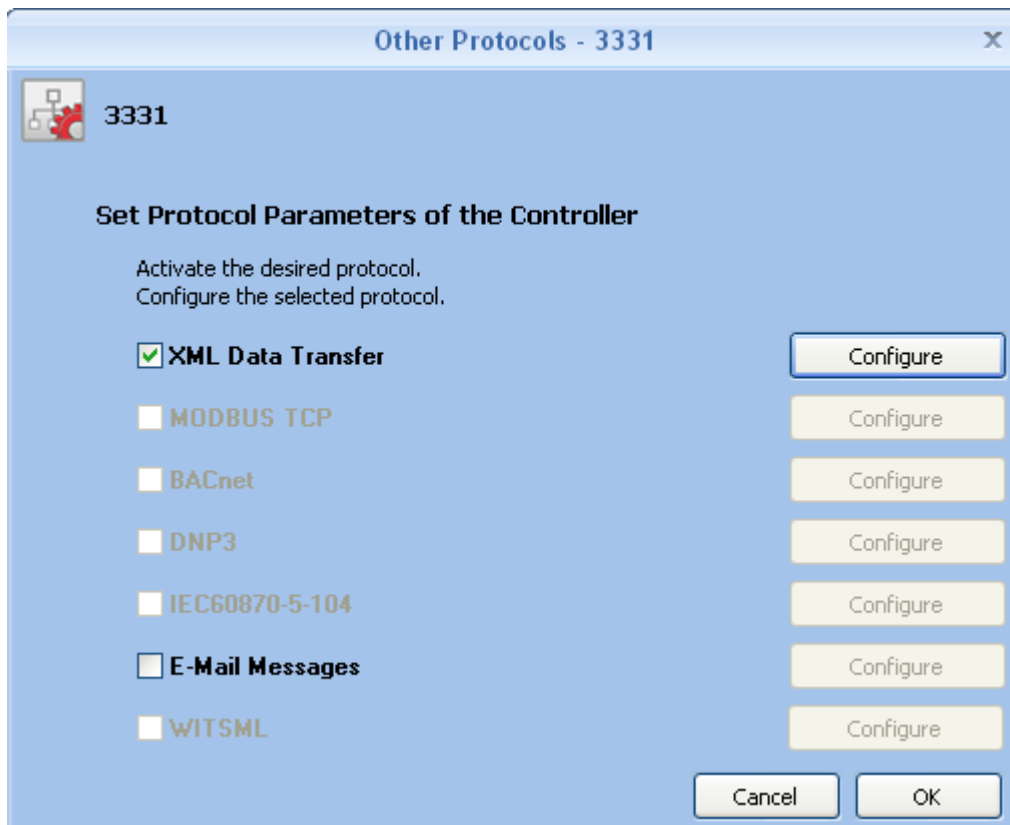


Figure 57: Activation of other protocols than LON using the OTHER PROTOCOLS window

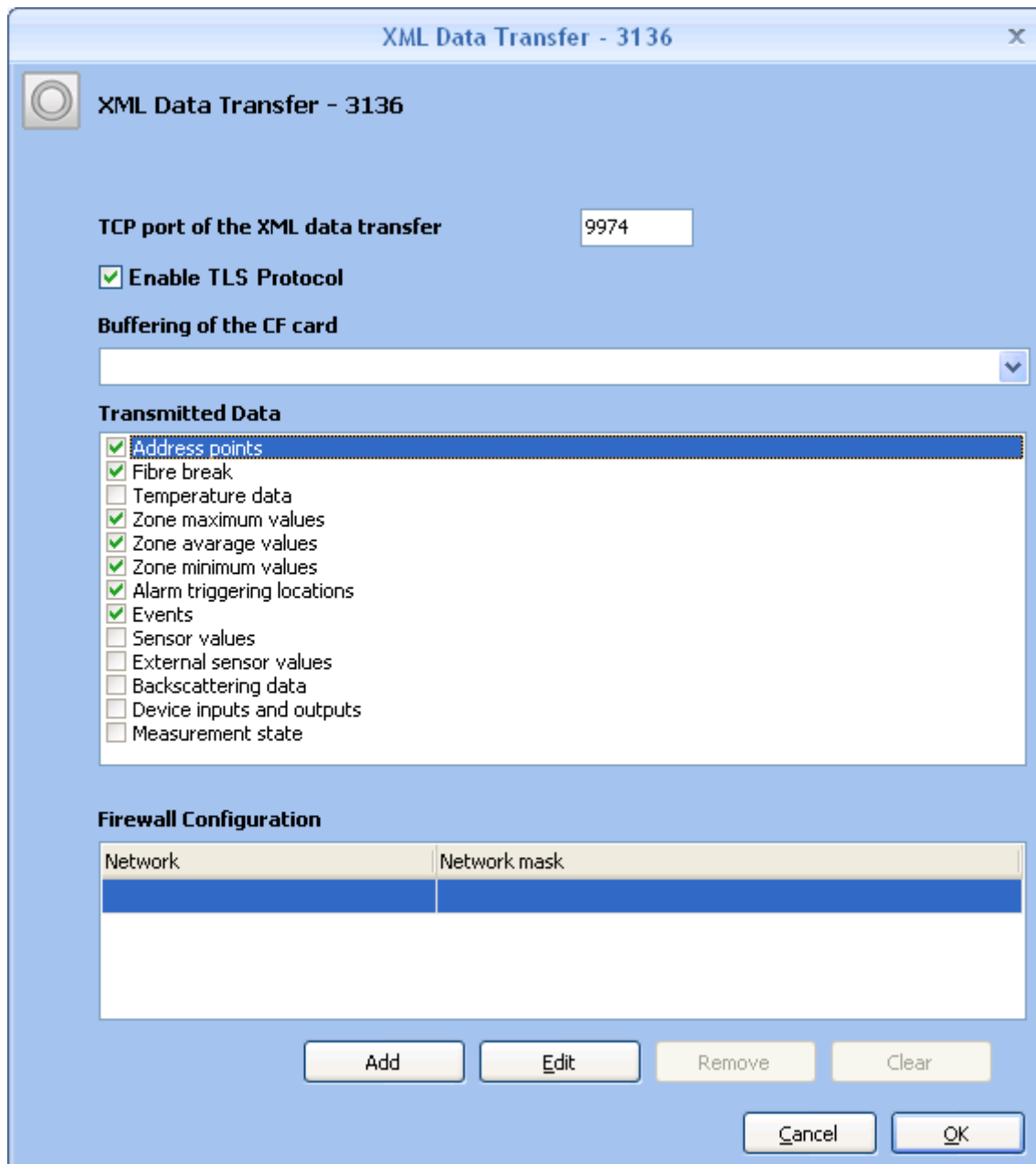


Figure 58: Configuration of the data transmission over the XML protocol



4.3.5 Locking the Controller

The controller can be locked using the LOCK CONTROLLER button of the COMMUNICATION menu of the DEVICE tab (Figure 59).

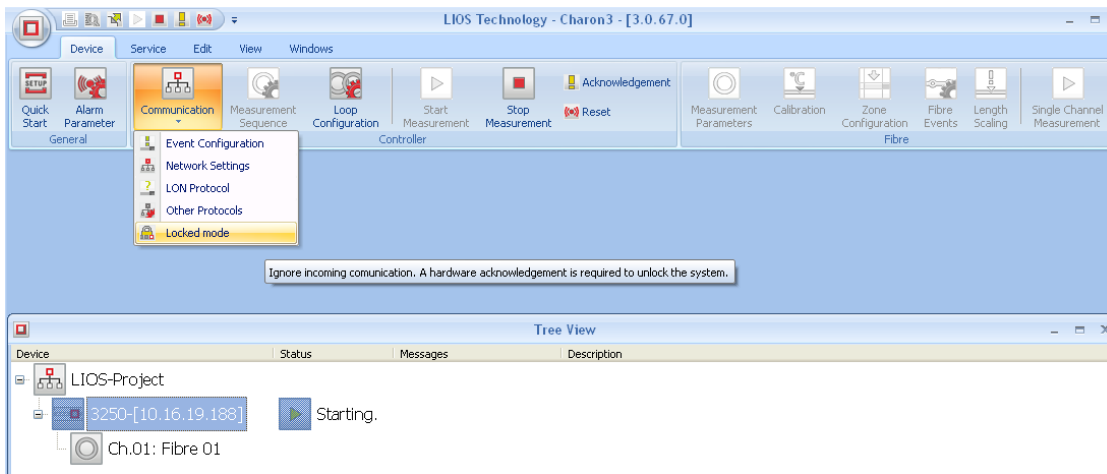


Figure 59: Locking the controller

A warning (Figure 60) must be confirmed to lock the controller. The locked state of the controller is indicated in the TREE VIEW by a locker icon in front of the controller name (Figure 61). The locked controller does not react to any software command send via communication interfaces. Thus, any unwanted unnoticed manipulation of the controller via the communication interfaces is avoided. The locked controller must be unlocked to again allow the access onto the controller. A hardware acknowledgement using the key switch or a correspondingly configured input is required to unlock the controller.

Caution!

According to the VdS approval, the controller must be locked to be compliant to EN 54-22. A locked controller cannot be unlocked by any software command.

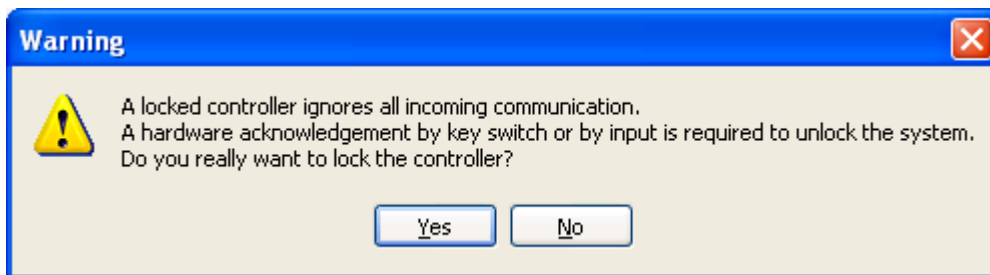


Figure 60: Warning before locking the controller

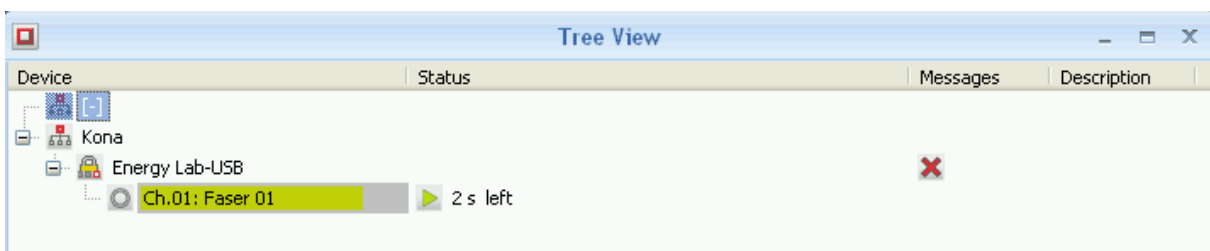


Figure 61: Display of a locked controller in the TREE VIEW (locker icon)



4.4 Measurement Sequence

The measurement sequence settings allow monitoring fibres in a user-defined sequence. Temperature profiles of each fibre are consecutively measured in the order as listed in the MEASUREMENT SEQUENCE field. The MEASUREMENT SEQUENCE window (Figure 62) can be accessed by pressing the MEASUREMENT SEQUENCE button in the DEVICE tab.

The measuring time of each fibre is given behind the names of the available fibres. In the column MEASUREMENT SEQUENCE, the measurement times are added up in the measurement sequence, and the cycle time is given behind the last fibre of the measurement sequence (Figure 62).

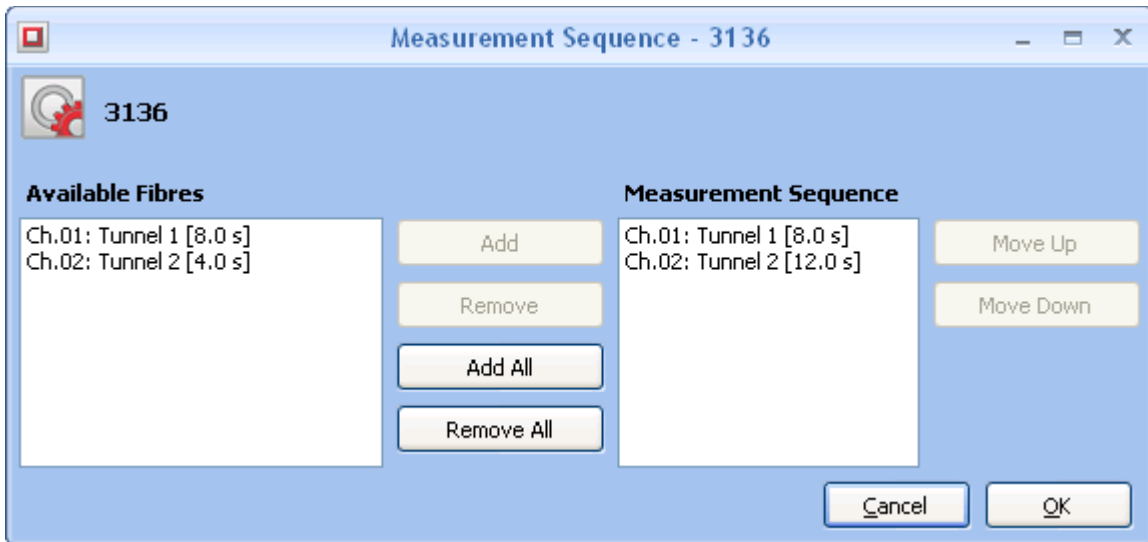


Figure 62: Measurement sequence window

In order to create a MEASUREMENT SEQUENCE, choose the fibre to be added and press the ADD button. Fibres selected in the MEASUREMENT SEQUENCE field can be removed by pressing the REMOVE button. In order to remove or add all available fibres the ADD ALL or REMOVE ALL buttons, respectively, can be used.

If more than one fibre is included in the measurement sequence, the sequence can be altered by selecting a fibre and pressing the MOVE UP / MOVE DOWN button. In order to establish frequent measurements on certain fibres, a fibre can be added multiple times.



For fire detection, the measurement sequence can only include one entry for each fibre. If there are multiple entries of the same fibre, the cycle time can be too long for compliance to the European, North American and national fire detection standards such as EN 54-22 and UL521 / ULC S530.



4.5 Fibre Settings

4.5.1 Measurement Parameters

The MEASUREMENT PARAMETER window (Figure 63) is available by selecting a fibre and clicking the MEASUREMENT PARAMETERS button in the DEVICE tab. It is also accessible by right clicking on the fibre and by choosing MEASUREMENT PARAMETER in the pull down menu.

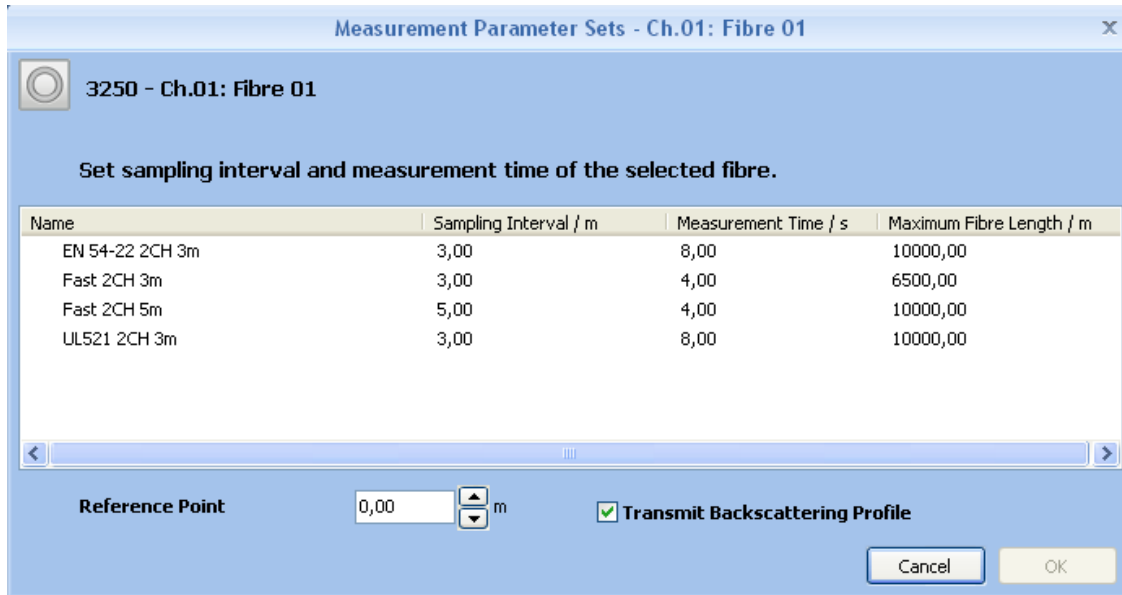


Figure 63: Measurement parameter window with pre-defined measurement parameter sets

The MEASUREMENT PARAMETER window shows the list of pre-defined measurement parameter sets available under consideration of the type of the controller, the number of channels and the length of the connected fibre. They are specified by their name, the sampling interval, the measuring time and the maximum fibre length.

The figure above shows an example of two approved measurement parameter sets “EN54-22 ...” and “UL521...”, and two non-approved measurement parameter sets “Fast ...” Further or other parameter sets are available for controllers with different numbers of channels and/or different lengths of fibre.



Since measurement parameters are crucial for fire detection response of the controller, only the approved parameter sets must be used to ensure operation compliant to European, North American and national standards such as EN 54-22 and UL521 / ULC S530. The use of any other measurement parameter sets is not compliant to the VdS and UL approval!

The measurement parameter sets are write-protected in order to ensure safe operation.

Select the proper measuring parameter set by a mouse click.

You can also enter the position of the reference point. The reference point is the distance from the controller connector to the zero position at the fibre optic cable. All zone definitions, alarm processing and other data outputs refer to the thus defined zero position. By default, the reference point is



set to 0 m meaning that the zero position is at the optical port of the controller. Confirm the selected parameters by pressing OK.

You can choose to transmit backscattering profiles by checking the right box. Backscattering profiles are useful for analysing the fibre quality in the sensor cable installation. They are not needed during permanent heat detection. Deactivate the transmission of backscattering profiles after commissioning in order to prevent unnecessary data transfer and loading of the database.

4.5.2 Temperature Calibration

The controllers are temperature-calibrated at the factory. Nevertheless, a re-calibration can be required during commissioning depending on the sensor cable installation. No re-calibration is required if the average temperature reading of the controller at a sensor cable position close to the end of fibre is within $\pm 3^{\circ}\text{C}$ of the reading of a point temperature reference at the same location.

The temperature calibration is available by selecting a fibre and by pressing the CALIBRATION button in the DEVICE tab, available when the controller is not measuring. It is also accessible by a right click on a fibre and by selecting CALIBRATION in the popup menu.

The temperature calibration automatically starts a warm-up and a calibration measurement. After finishing the calibration measurement, the TEMPERATURE CALIBRATION window opens (Figure 64).

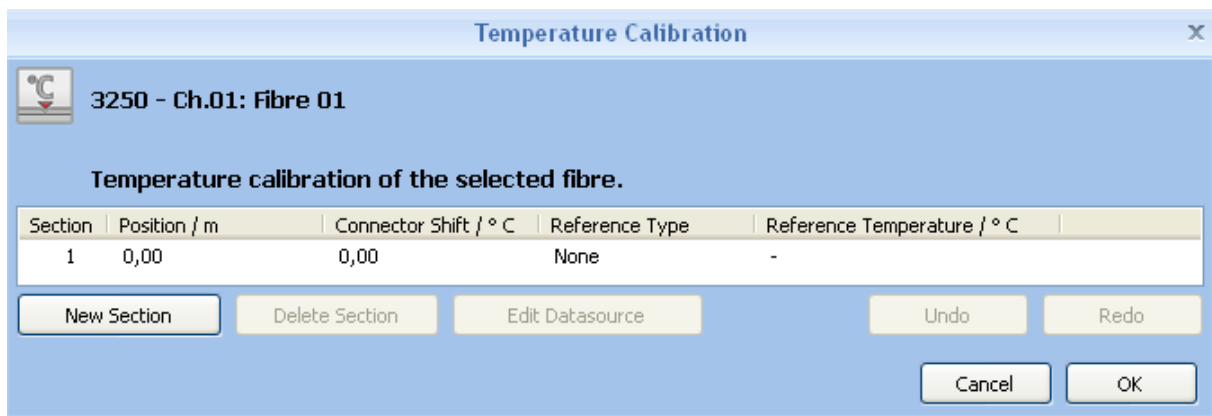


Figure 64: Temperature Calibration window

By default, only the single section 1 covering the whole fibre length is defined. Additional sections are only required if there are temperature shifts at connector or splice positions or if there is an uneven differential attenuation along the fibre. The latter situation is indicated by different unrealistic temperature slopes in different sections of the fibre. The fibre length can be split into multiple sections using the NEW button in the TEMPERATURE CALIBRATION window. After defining all sections, each section needs to be calibrated individually.

Section 1 starts at connector position 0 m (connector of controller, not the reference point!). If you see an unrealistic shift of the temperature reading at the connector position, it can be corrected by entering the value in the CONNECTOR SHIFT / °C column. Please check the correct reading of the temperature behind the connector position using a calibrated reference temperature sensor. Adjust the value of the CONNECTOR SHIFT until readings of controller and reference agree within $\pm 1^{\circ}\text{C}$.



The reference temperature is the temperature value at the end of the corresponding fibre section. If the average temperature reading of the controller at this position deviates by more than +/-3°C from that of a calibrated reference sensor at the same position, enter the reading of the reference sensor in the REFERENCE SENSOR/ °C column. Adjust the value of the REFERENCE SENSOR until readings of controller and reference agree within +/- 2°C.

The fibre length can be split into multiple sections using the NEW button in the TEMPERATURE CALIBRATION window. After defining all sections, each section needs to be individually calibrated in the TEMPERATURE CALIBRATION window. Reference temperature values must be determined before and after each connector position. Data needs to be entered in the sequence from the lowest to the highest section number following the same procedure as given above for section 1. Nevertheless, we strongly recommend improving the quality of the fibre installation by using proper splices and fibres instead of section-wise calibration.

4.5.3 Zone Configuration

In order to define or modify zones, select the fibre of interest and press the ZONE CONFIGURATION button in the DEVICE TAB or right click on the fibre and press ZONE CONFIGURATION. The ALARM ZONES window opens (Figure 65).

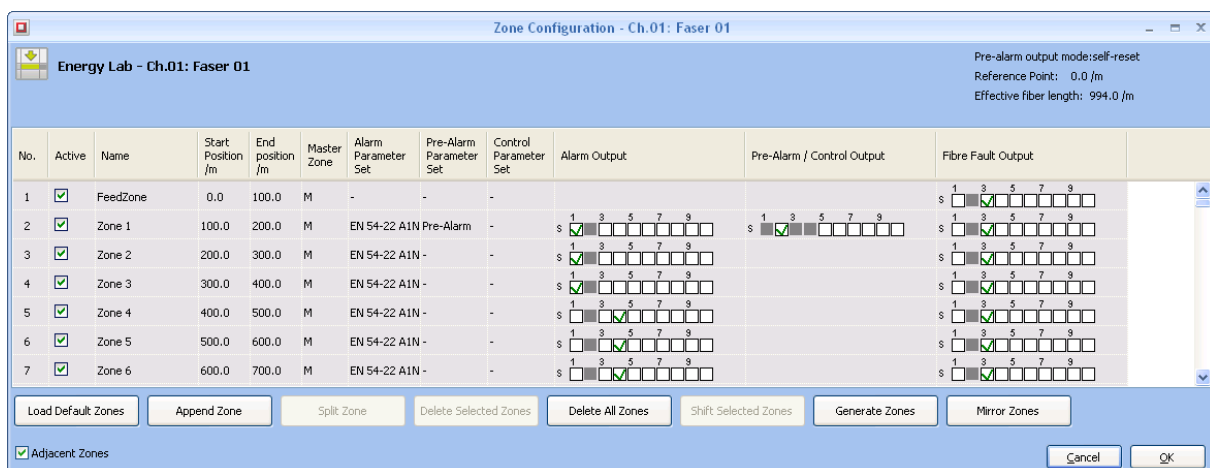


Figure 65: Zone configuration window

The button LOAD DEFAULT ZONES allows the user to load a set of pre-defined zone definitions from an application setting (see section 7.3.2.1). The zone positions, trigger parameter sets and outputs are automatically set for the whole fibre.

You can edit all zone settings except the number shown in the left column by double clicking the respective fields or using the F2 function key. You can select single or multiple zones for modification using the left mouse, SHIFT and CTRL keys.

If the ADJECENT ZONES box is activated, any end position of a zone that is edited will be automatically used as start position of the following zone and vice versa.

You can append a new zone after the last zone by pressing the APPEND ZONE button. A warning is show if the last zone has already reached the end of fibre, and no new zone is generated in this case.

You can split a zone into two zones with half the length by selecting a zone and pressing the SPLIT ZONE button.



You can delete the selected or all zones by pressing the buttons DELETE SELECTED ZONES or DELETE ALL ZONES.

You can shift selected zones by pressing the button SHIFT SELECTED ZONES. The SHIFT SELECTED ZONES window opens (Figure 66). Enter the desired shift distance in meters and confirm the shift by pressing OK.

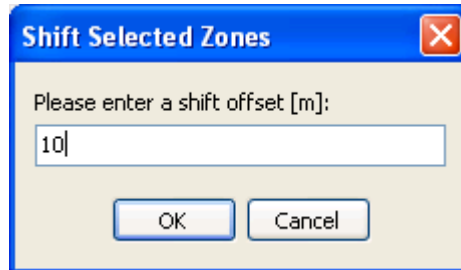


Figure 66: Shift selected zones window

You can generate single or multiple zones by pressing the GENERATE ZONES button. The GENERATE ZONES window opens (Figure 67). You can edit the start and end positions, the length of zones or the number of zones, and you can select alarm and pre-alarm parameter sets for the zones. You can also generate a master zone with slave zones by activating the FIRST ZONE HAS SLAVE ZONES checkbox.

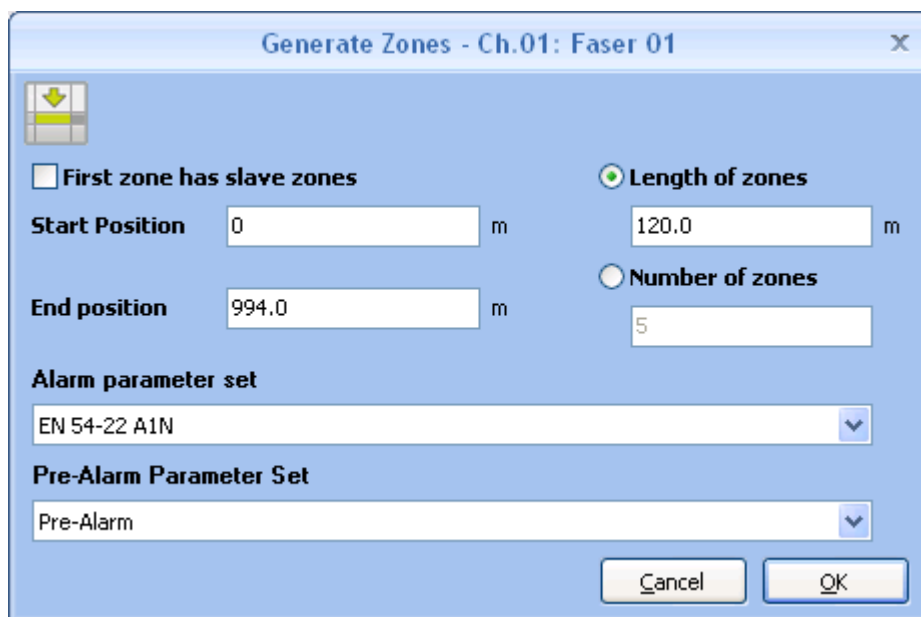


Figure 67: Generate zones window

The order of zones along a sensor cable is reversed by using the button MIRROR ZONES. This function is useful when you measure two fibres of a sensor cable with two controllers from opposite ends (redundant operation).

Zones are (de)activated by (un)checking the check box in the ACTIVE column. Deactivated zones do not trigger alarms or other events.

The name of a zone is edited by double clicking on the respective cell in the NAME column or by pressing F2 after selecting the entry. Start and end positions of a zone are edited in the same way using the START and END columns.



By default, zones are defined as master zones indicated by an M in the column MASTER ZONE. Trigger parameter sets and outputs can be assigned individually for each master zone.

Zones can be changed to child zones by entering the number of a master zone into the column MASTER ZONE. Then, trigger parameters and outputs are identical to that of the master zone. Only the start and end position as well as the number of the master zone need to be entered to define a child zone completely. This concept enables an efficient configuration of large numbers of zones using identical triggering parameters.

Choose the alarm parameter set for each alarm zone using the ALARM PARAMETER SET column. Alarm parameter sets approved for different response classes according to EN 54-22 are supplied with the controller. The use of other alarm parameter sets is not compliant to the approval.

You can choose a pre-alarm parameter set for each zone using the PRE-ALARM PARAMETER SET column. Pre-alarm parameter sets include the same types of parameters as alarm parameter sets. Usually, pre-alarm parameter sets are set to a higher sensitivity to heating or fires. Compliance to standards is not required for pre-alarm parameter sets.

Alternatively, you can choose a control parameter set for each zone using the CONTROL PARAMETER SET column. It is not possible to assign both, a pre-alarm and a control function to a zone.

If the activation temperature of the control parameter set is higher than the deactivation temperature, the assigned output is activated if one location within the zone has a temperature above the activation temperature. The output is deactivated if all locations within the zone have temperatures below the deactivation temperature. If the activation temperature of the control parameter set is lower than the deactivation temperature, activation and deactivation of the output is triggered by temperatures below the activation and deactivation temperatures.

Programmable output(s) can be assigned to a zone by left-clicking on the respective check box in the ALARM OUTPUT, PRE-ALARM OUTPUT and FIBRE FAULT OUTPUT columns. The output COLLECTIVE ALARM is set independently from the choice of programmable outputs as alarm outputs, when an alarm criterion is met. Since alarm and fibre break outputs are manual reset, an output assigned to an alarm or fibre break is no longer available for a pre-alarm if pre-alarms are self-reset. If the control function is selected, the pre-alarm / control outputs are assigned to the control function. Control outputs are always self-reset.

4.5.4 Fibre Events

In order to define or modify fibre events (fire magnitude and direction), select the fibre of interest and press the FIBRE EVENTS button in the DEVICE TAB or right click on the fibre and press FIBRE EVENTS. The FIBRE EVENTS window opens (Figure 68).

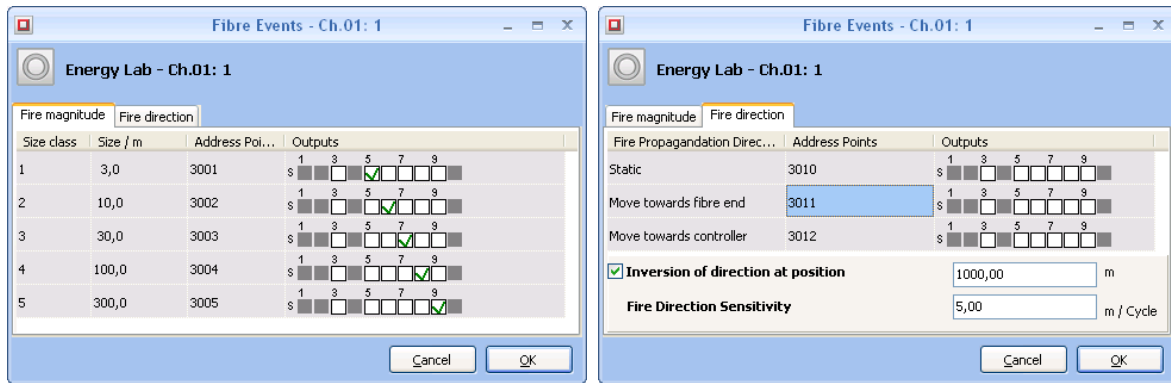


Figure 68: Fibre events window

The FIRE MAGNITUDE tab (Figure 68, left) shows the five fire magnitude classes. The size is the total distance range of adjacent alarm triggering locations. You can change the sizes by double clicking into the field and entering a new number. You can assign self-reset outputs to fire magnitude classes by checking boxes in the OUTPUTS column.

The FIRE DIRECTION tab (Figure 68, right) shows the three fire directions static, moving towards fibre end and moving towards controller. You can assign self-reset outputs to fire directions by checking boxes in the OUTPUTS column. If activated by checking the respective box, the INVERSION OF DIRECTION AT POSITION entry determines a distance point where the directions are reversed. This feature allows changing the assignment of fire directions for installations with a loop-like sensor cable laying. The FIRE DIRECTION SENSITIVITY entry determines the speed limit for considering a fire as moving.

4.5.5 Length Scaling

For special sensor cable layouts the length scaling feature allows the adjustment of the visualized length of the temperature profile to the object to be monitored. For example, if the sensor cable is wound around an object (e.g. a pipe) the sensor cable is longer than the length of the monitored object.

In order to adjust the length of visualized temperature profile to the length of the object, select the fibre in the tree view and press the LENGTH SCALING button in the DEVICE TAB or right click on the fibre and press LENGTH SCALING. The LENGTH SCALING window opens (Figure 69) for the configuration which part of the sensor cable shall be scaled.

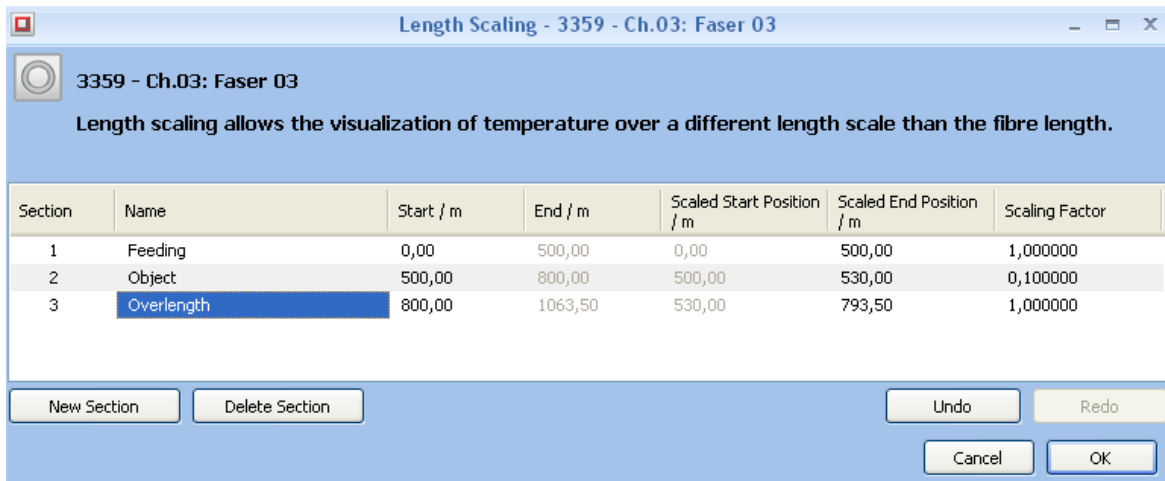


Figure 69: Length scaling window

In Figure 69 there are three sections specified. The 1st section has a neutral scaling factor of 1 and is 500m long. This section may be a feeding line. The 2nd section of the sensor cable has a length of 300m. The scaling factor of this section is set to 0.1 considering e.g. a meander shaped layout of the sensor cable on the object, which is 10 times smaller. The 3rd section has a neutral scaling factor again.

The definition of this three sections results in a temperature profile as shown in Figure 70.

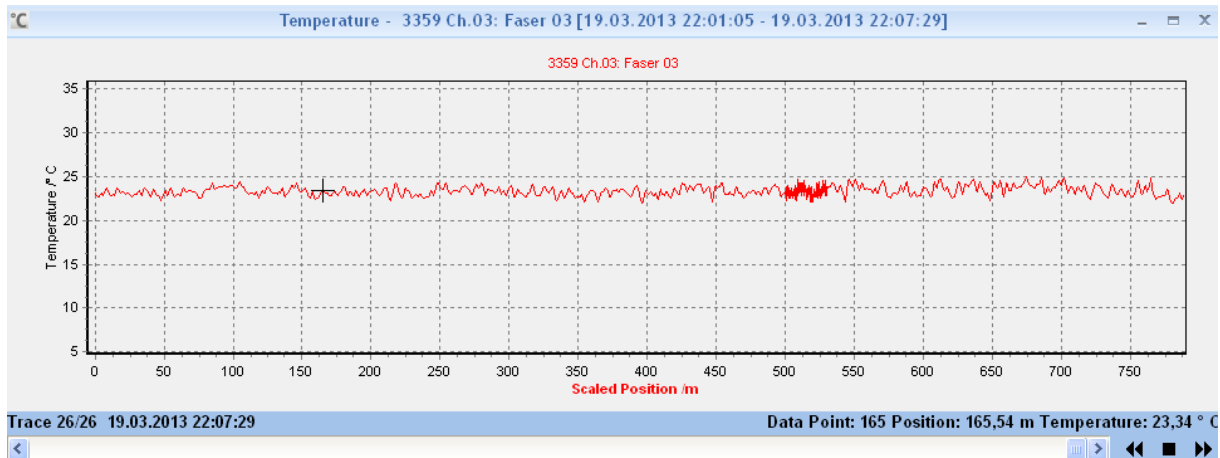


Figure 70: Temperature profile with the scaled section 2

The 300m long sensor cable section beginning at position 500m has a scaling factor of 0.1, which results in a 30m long section in the temperature profile.

4.6 Commissioning Using Different Charon3 Instances

During commissioning, the controller configuration is stored in the controller memory and in the Charon3 database. If you use different computers for commissioning a controller or if you commission a controller offline, the configuration in a Charon3 database can differ from the one on the controller if the controller was last commissioned using another Charon3 instance. In this case, the CONFIGURATION ON CONTROLLER AND PC ARE DIFFERENT window (Figure 71) opens after connecting Charon3 to the controller.



You can download the configuration from the controller to the Charon3 database by pressing the **CONTROLLER CONFIGURATION** button. The configuration is downloaded automatically after 30 s if you do not press any button. The configuration on the PC is replaced by downloading the configuration from the controller.

You can upload the configuration from the Charon3 database to the controller by pressing the **PC CONFIGURATION** button, or you can disconnect from the controller by pressing **DISCONNECT**.

We recommend downloading configurations from the controller if there is no specific reason for an upload since the configuration on the controller is the latest one.

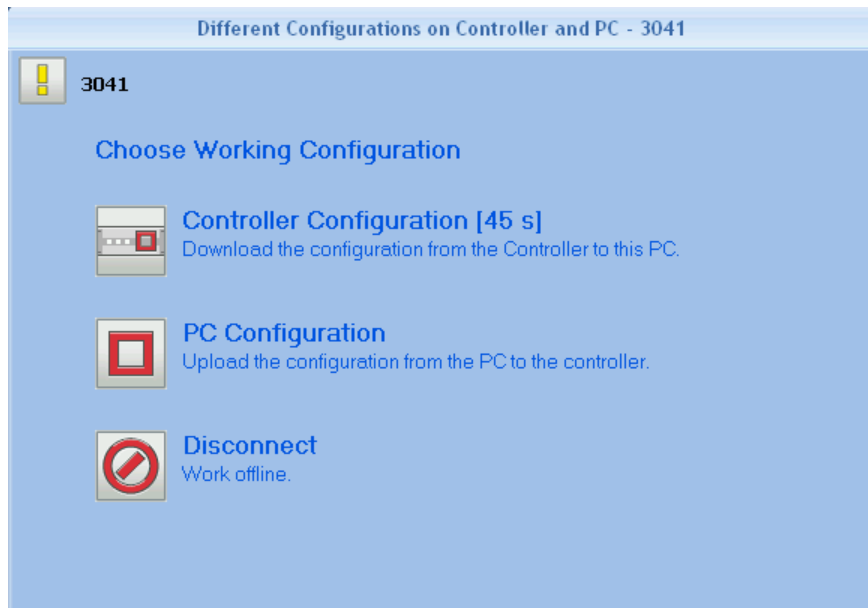


Figure 71: Different configurations on controller and PC window



5 Operating the Controller

Temperature data can be taken for all channels included in the measurement sequence or for a single channel using buttons on the DEVICE tab (Figure 72).

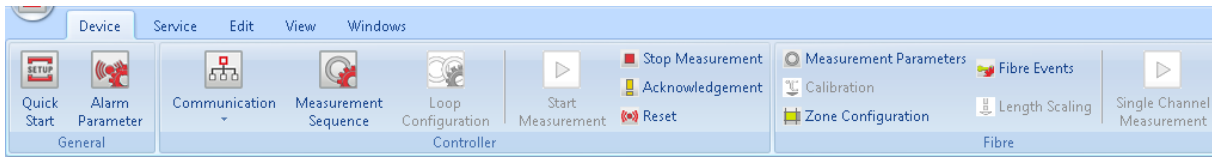


Figure 72: Task bar of the DEVICE tab

5.1 Measurement

In order to start a continuous acquisition of temperature profiles in a measuring sequence of a specific controller, mark the respective controller in the tree view and press the START MEASUREMENT button (Figure 72). Alternatively, a measurement is started by right clicking on the controller and by selecting START MEASUREMENT in the pull down menu (Figure 26).

After starting a measurement, the controller is warmed-up for about one measurement cycle time. The warm-up progress is indicated in the tree view (

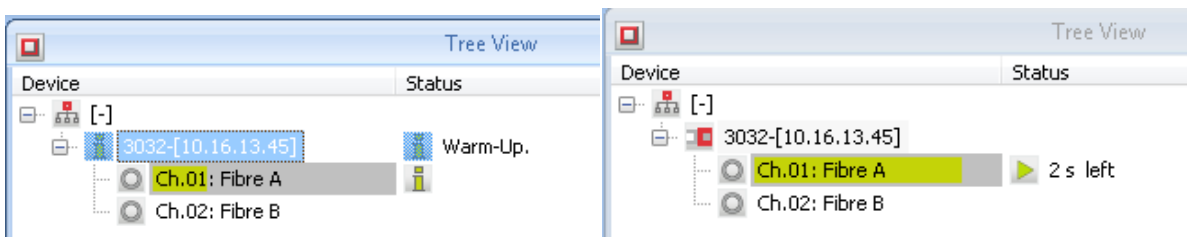


Figure 73: Tree view during warm-up (left) and running measurement (right)

After finishing the first measurement cycle, a temperature profile window (7.4.1) opens showing the measurement result. Backscattering profile (7.4.2) and zone view (7.4.1) windows also open if the transmission of backscattering profiles is activated (4.5.1) and if zones are defined and active (4.5.3), respectively.

The measurement sequence is continuously repeated until the STOP MEASUREMENT button is pressed. Alternatively, a measurement can be stopped by right clicking on the controller and by selecting STOP MEASUREMENT in the pull down menu (Figure 26).

5.2 Single Channel Measurement

In order to start a continuous temperature measurement of a specific channel, either right click on the respective fibre within the tree view and select SINGLE CHANNEL MEASUREMENT in the pull down menu or mark the fibre of interest and press the SINGLE CHANNEL MEASUREMENT button in the FIBRE section of the DEVICE tab. Press the STOP MEASUREMENT button to stop the data acquisition.

5.3 Acknowledgement and Reset

Acknowledgement and reset are triggered using the corresponding buttons in the DEVICE tab or using correspondingly configured inputs (section 4.3.1).



5.4 Evaluation of Sensor Fibres Using Backscattering Profiles

You can evaluate the quality of a sensor fibre using its attenuation coefficient, the magnitude of point losses and the reflection from the optical termination of the fibre end. The transmitting of backscattering profiles is activated as described in section 4.5.1. The evaluation of these parameters is discussed within this section.

All mentioned parameters are derived from backscattering profile (7.4.2). Select the STOKES channel as well as the LOGARITHMIC view in the right upper corner of the BACKSCATTERING window.

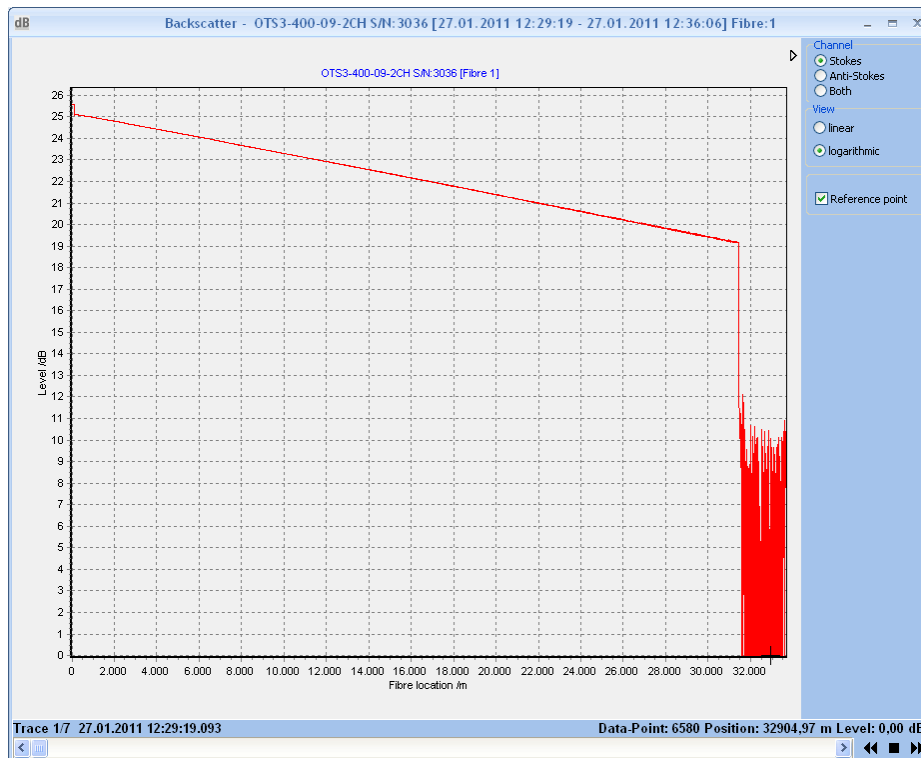


Figure 74: BACKSCATTERING window

5.4.1 Attenuation

The attenuation coefficient represents the attenuation of the optical power along the fibre. Its unit is dB/km. Excessive attenuation can result in poor measurement quality. It can be caused by mechanical stress such as crushing, bending or tension forces.

In order to determine the attenuation coefficient, double click on the backscattering profile right after the reference (zero) position (see the bold cross at the left side of Figure 75) and move the mouse cursor close to the end of the fibre. The attenuation coefficient is then displayed as ATTENUATION at the bottom right corner of the BACKSCATTERING window. An attenuation below 0.4 dB/km is recommended for proper measurement performance for LHD3 controller types. For LHD3S controller types an appropriate attenuation verified in the Stokes channel is below 1.2 dB/km.

The false alarm safety for linear heat detection primarily depends on the noise level of the temperature measurement and the alarm parameterisation. The noise level of the temperature measurement primarily depends on the optical fibre properties (fibre attenuation and point losses, s.a. chapter 5.4.2) and the measurement parameters. The approved combinations of measuring parameters and alarm parameters for a dedicated maximum fibre length shown in chapter 9 base on a the above specified maximum levels of the attenuation.



Higher attenuation coefficients can be acceptable for short fibre lengths.

Please contact the LIOS support in case of higher attenuations.

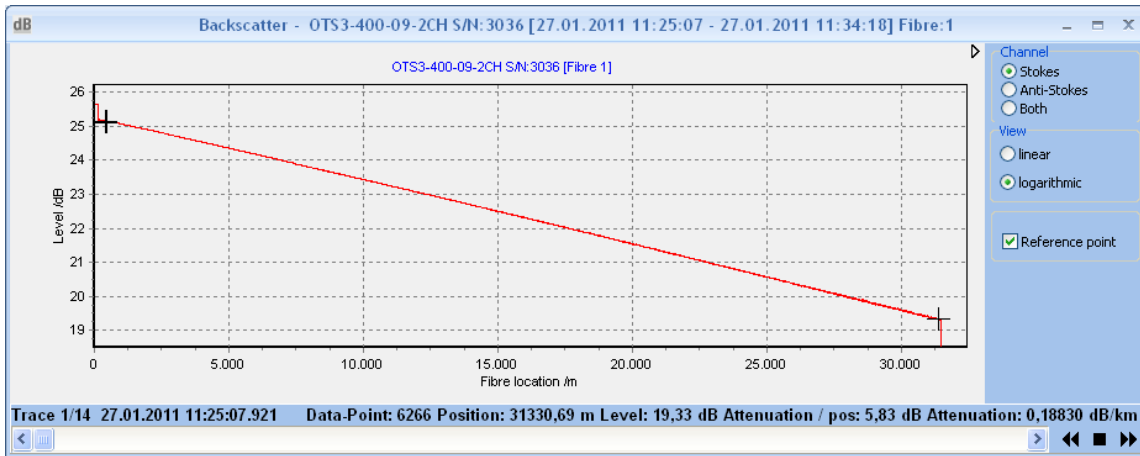


Figure 75: BACKSCATTERING window: Determination of the attenuation coefficient

5.4.2 Point losses

Splices and connectors introduce point losses (localised attenuations). Excessive point losses may affect the measuring performance. The point loss is determined by double clicking on the backscattering profile directly before the splice or connector and by moving the mouse cursor to a position right after the splice. The attenuation is then shown below the BACKSCATTERING as ATTENUATION / POS. Splices with an attenuation exceeding 0.1 dB need to be reworked in order to ensure proper measuring performance. Connectors other than the one at the controller should be avoided completely.

5.4.3 Termination

The end of the fibre must be optically terminated in order to ensure proper measuring performance. Strong reflections from a non-terminated end of fibre would affect temperature readings close to the end of fibre.

The backscatter curve of a proper (left) and an improper (right) optical termination is shown in Figure 76. In order to evaluate the reflection, double click on the backscatter trace right before the reflection and move the mouse cursor to its peak position. The reflection factor is then shown below the BACKSCATTERING as ATTENUATION / POS. (Disregard the minus sign.) The magnitude of an optical reflection depends on the setting of the sampling interval. The optical reflection is resolved more clearly and its magnitude becomes greater, if a smaller sampling interval is used. Optical terminations with reflections larger than 0.5 dB need to be reworked in order to ensure proper measuring performance.

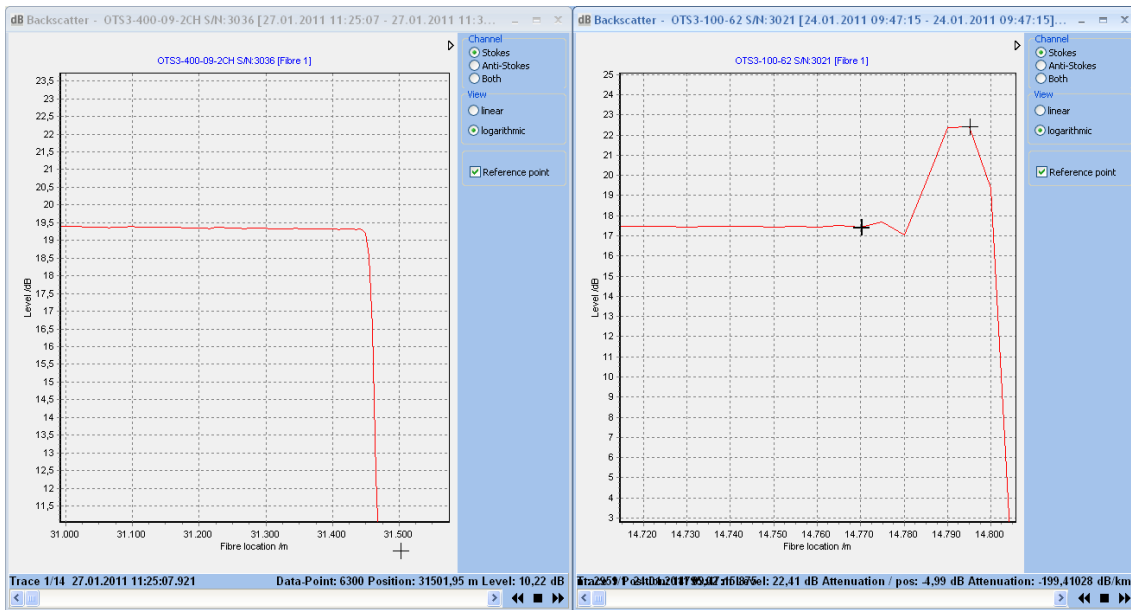


Figure 76: BACKSCATTERING windows showing curves indicating proper (left) and improper (right) optical termination of the fibre.



6 Servicing the Controller

The controller can be serviced using the commands shown in the SERVICE tab (Figure 77).

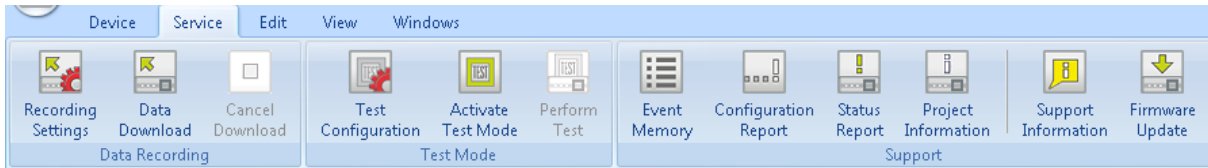


Figure 77: Service tab of Charon3

6.1 Data Recording

Data recording on the controller and download of recorded data can be controlled using the DATA RECORDING section of the SERVICE tab (Figure 77).

6.1.1 Recording Settings

The RECORDING SETTINGS window (Figure 78) opens after pressing the RECORDING SETTINGS button. You can select one of the two recording modes: DISABLED and TRIGGERED BY ALARMS. The setting CONTINUOUSLY is not available for LHD3-series controllers.

No data is recorded in the DISABLED mode. If the TRIGGERED BY ALARMS mode is selected, measurement data is recorded for one hour before and one hour after an alarm.

Caution!

About 4 bytes of memory are required to record a data point. A single temperature profile may consist of thousands of data points, thus requiring several kilobytes of memory. Download recorded data regularly and delete downloaded data recordings on the controller. Otherwise data may be lost.

You can select two different storage modes: RING BUFFER and LINEAR BUFFER (Figure 78). In the RING BUFFER mode, older data is overwritten with new ones when the memory card is filled by 80% with measurement data⁴. In the LINEAR BUFFER mode, the data recording stops when the memory card is filled by 80% with measurement data.

Press OK to confirm the RECORDING SETTINGS.

⁴ The remaining 20% are reserved for the event memory of the controller.

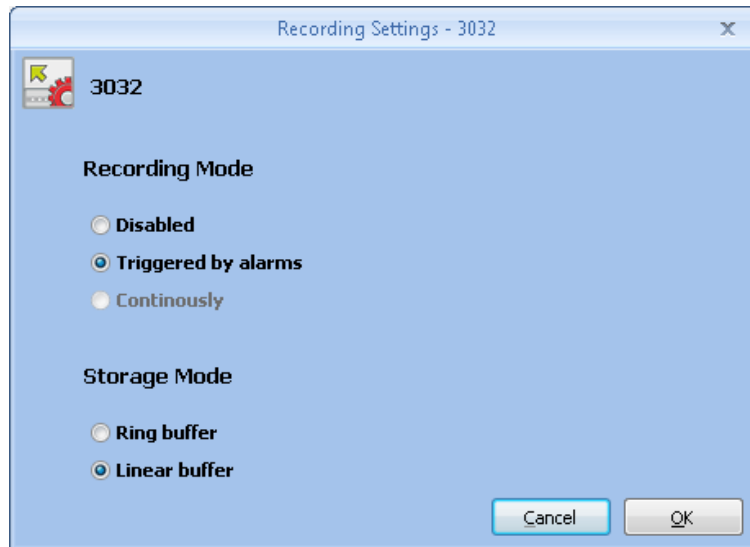


Figure 78: Recording settings window

6.1.2 Data Download

Press the DATA DOWNLOAD button of the SERVICE tab (Figure 77) to download data recordings from the controller. The DOWNLOAD DATA window (Figure 79) opens and shows the number of data recordings of the controller. By checking the box, you can select to automatically delete data recordings stored on the controller after downloading. We recommend choosing this option to release storage capacity for future data recordings. Choose DATA DOWNLOAD for downloading all data recordings, DELETE ALL for deleting all data recordings without downloading or just CLOSE to close the window without further action.

Downloaded data are automatically saved as new Measurement Session in the Measurement Explorer. Depending on the number of data recordings and transmission speed, downloading the data recordings may take several minutes or even an hour. The status bar of the controller (Figure 80) in the TREE VIEW indicates the progress of the download. You may cancel the download by pressing the CANCEL button of the SERVICE tab (Figure 80).

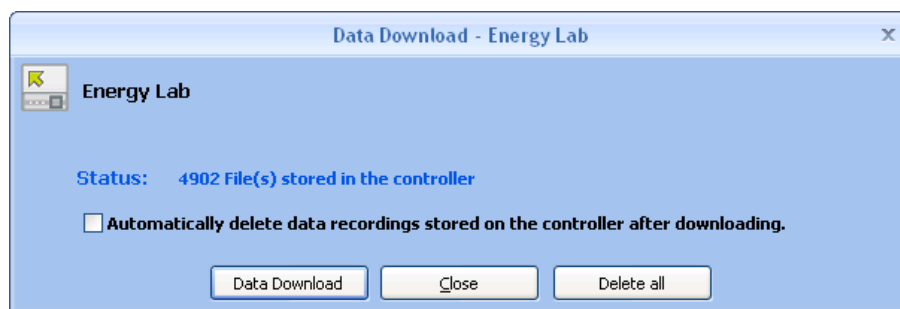


Figure 79: Data download window

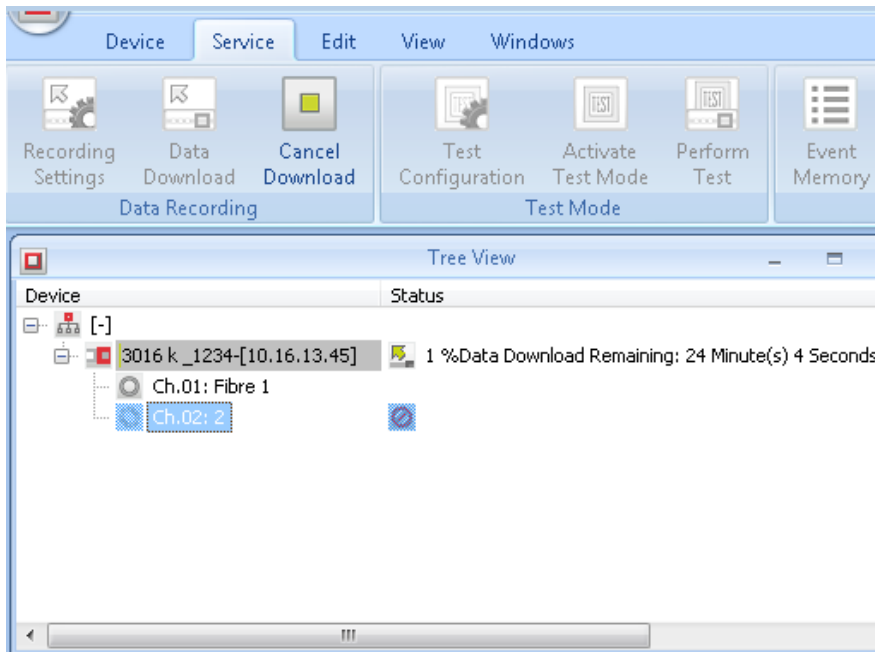


Figure 80: CANCEL DOWNLOAD button and TREE VIEW window during download of data

6.2 Test Mode

6.2.1 Test Mode Settings

Press the TEST CONFIGURATION button in the SERVICE tab in order to modify the test mode settings of the controller. The TEST MODE SETTINGS window opens (Figure 81).

You can enable and select a test mode alarm parameter set. As a result, the selected parameter set is used as unitary alarm parameter set for all zones defined for the controller. This allows you to adapt the alarm triggering parameters to fit to the equipment used for generating test heat events at the sensor cable. If the test mode alarm parameter set is not enabled, the individual alarm parameter sets configured for each of the zones are used in the test mode as well.

Alarm outputs, buzzer operation, alarm transmission via interfaces, transmission of temperature data as well as acknowledgement and reset notification can be disabled in the test mode by checking the respective boxes in the TEST MODE SETTINGS window.

Close the window by pressing OK if the selected settings shall be saved. Otherwise, press CANCEL.

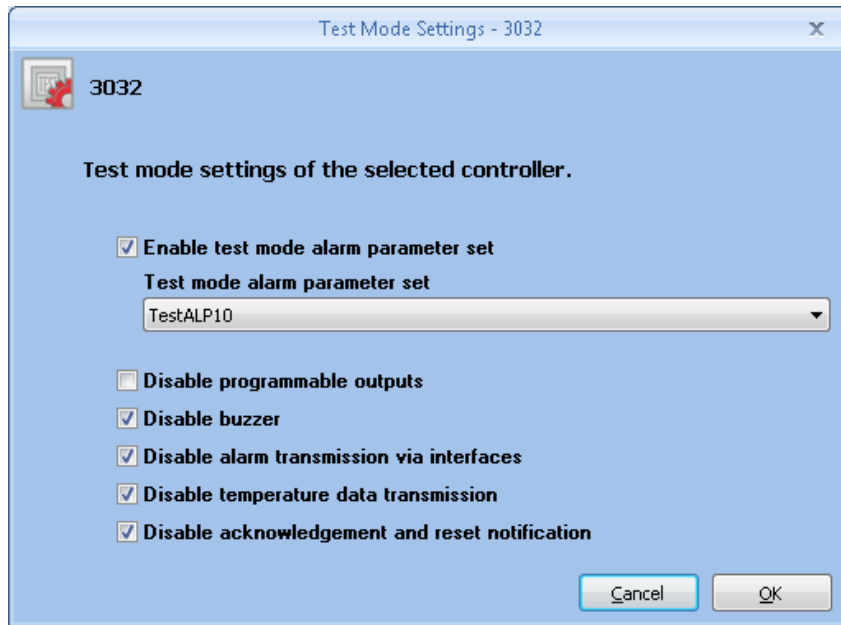


Figure 81: Service: Test mode settings

6.2.2 Activation and Deactivation of the Test Mode

The test mode is activated by pressing the ACTIVATE TEST MODE button in the SERVICE tab (Figure 77). You have to confirm the activation of the test mode by pressing YES in the confirmation window. Then, the controller is operated in the test mode. The controller status displayed in the TREE VIEW changes to TEST MODE (Figure 82).

Caution!

The collective fault output of the controller is activated, and respective address points are sent. Test mode settings are used.

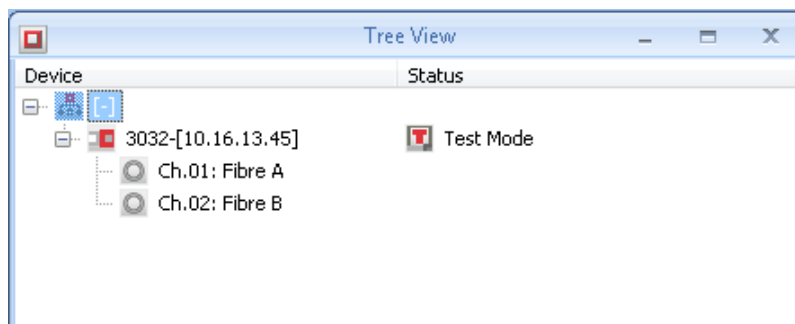


Figure 82: Tree view of a controller in the test mode

The middle button in the TEST MODE section of the SERVICE tab is then renamed to DEACTIVATE TEST MODE (Figure 83). Press this button to return the controller to the normal operation mode. You have to confirm the deactivation of the test mode by pressing YES in the confirmation window.

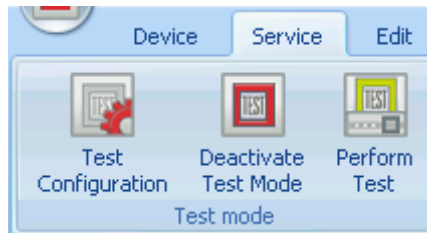


Figure 83: Test mode section of the service tab after activation of the test mode.

6.2.3 Performing Controller Tests

In the test mode, the controller is fully operational, but with the modifications mentioned above. The test mode can be used to trigger test alarms using heating equipment for the sensor cable.

Certain events can be simulated in the test mode after pressing the PERFORM TEST button of the SERVICE tab. Then, the PERFORM TEST window (Figure 84) shows the outputs, inputs and fibres of the controller. Events on individual or all outputs, inputs or fibres are simulated after selecting the respective hardware entry by double clicking the entry or by pressing the ACTIVATE / DEACTIVATE buttons.

Simulation of an active output means that the output is activated.

Simulation of an active input means that all actions (alarm, output, address point) configured for activation of the input are executed.

Simulation of an active fibre means that all alarms and fibre breaks configured for any of the zones are triggered.

Simulation of an active zone means that the alarm and fibre break configured for the zone are triggered.

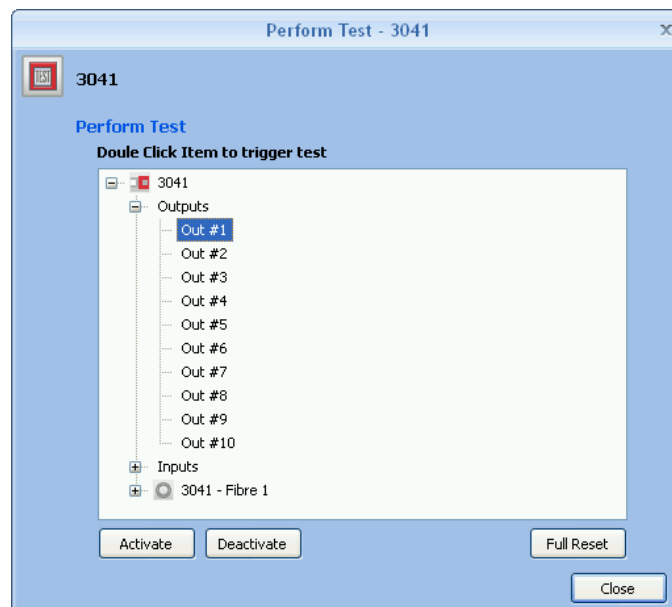


Figure 84: Perform test window

The controller can be returned to a non-activated test state by pressing the FULL RESET button. A FULL RESET is a RESET plus deactivation of all inputs and self-reset outputs. Finish the tests by pressing the CLOSE button.



6.3 Support

6.3.1 Event Memory

The controller has a permanent event memory. Any events of the controller such as alarms, faults, warnings and further messages are saved until the capacity limit of the memory is reached. Afterwards, the older entries are overwritten.

The content of the event memory of the controller is downloaded, saved in the Charon3 database and displayed after pressing the EVENT MEMORY button in the SERVICE tab (Figure 85). The list of events consists of several pages if the buttons NEXT and BACK can be selected. The period of time of each page of the event entries is shown on the top left of the window EVENT MEMORY. The list of events can be filtered by type of event, source and/or time by checking the boxes in the FILTER column. The numbers in brackets behind the words FILTER OPTIONS indicate the number of filtered events and the total number of events. The list can be also arranged by time, type, source or description in ascending or descending order by clicking the headlines of the respective columns.

You can save the event memory to a rtf file or print it using the SAVE and PRINT buttons on the lower right corner of the window.

The content of the event memory can be deleted from the Charon3 database or from the controller by pressing the respective buttons in the upper right corner of the EVENT MEMORY window. If you delete the events from the database only, the events remain in the controller memory and are downloaded again after the next opening of the EVENT MEMORY window. If you delete the events from the controller memory only, the events remain in the Charon3 database and are further visible in the EVENT MEMORY window, but they are no more saved on the controller. These events are then not available for another PC connected to the controller.

We recommend downloading the event memory during any service and deleting the event memory of the controller afterwards.

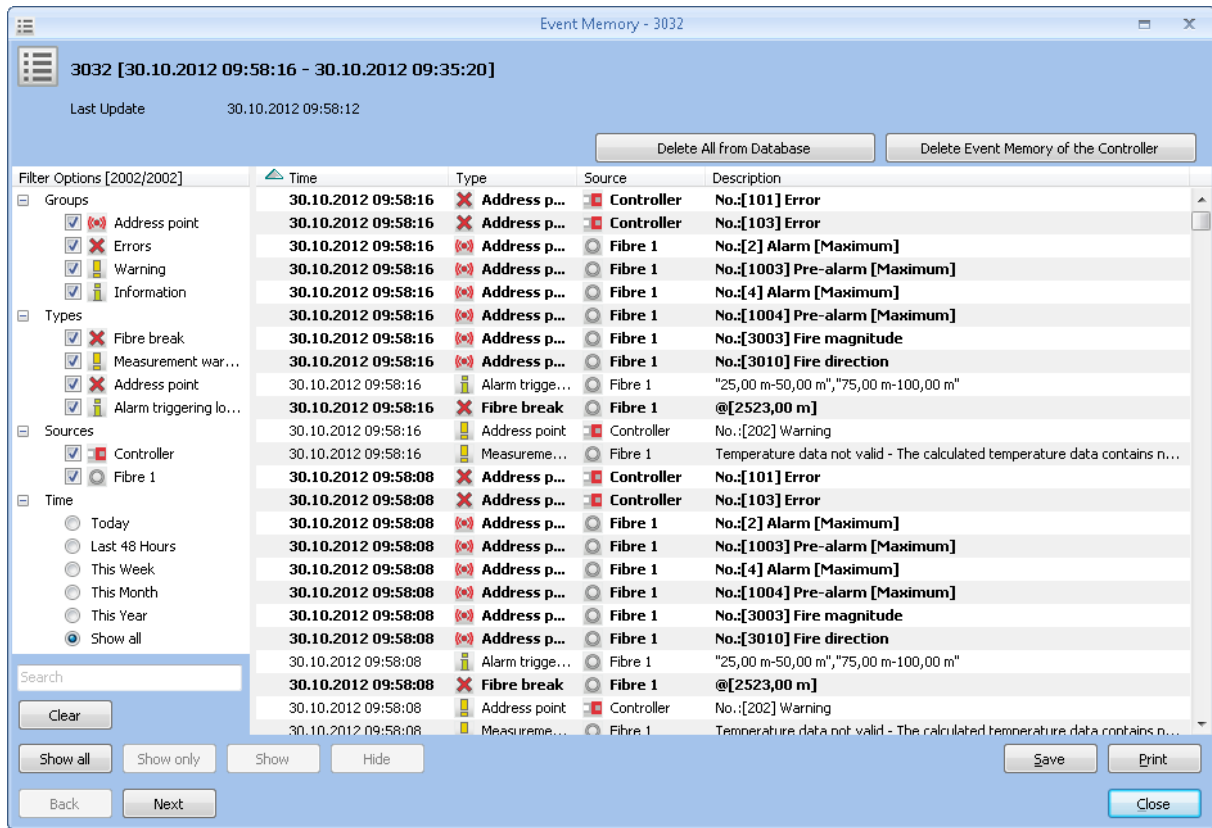


Figure 85: Event memory window

6.3.2 Configuration Report

A controller configuration report is generated automatically by pressing the CONFIGURATION REPORT button of the SERVICE tab. The report includes controller and fibre settings. A preview is shown in the PREVIEW window (Figure 86). The report can be saved or printed using the respective buttons in the upper left corner of the PREVIEW window.

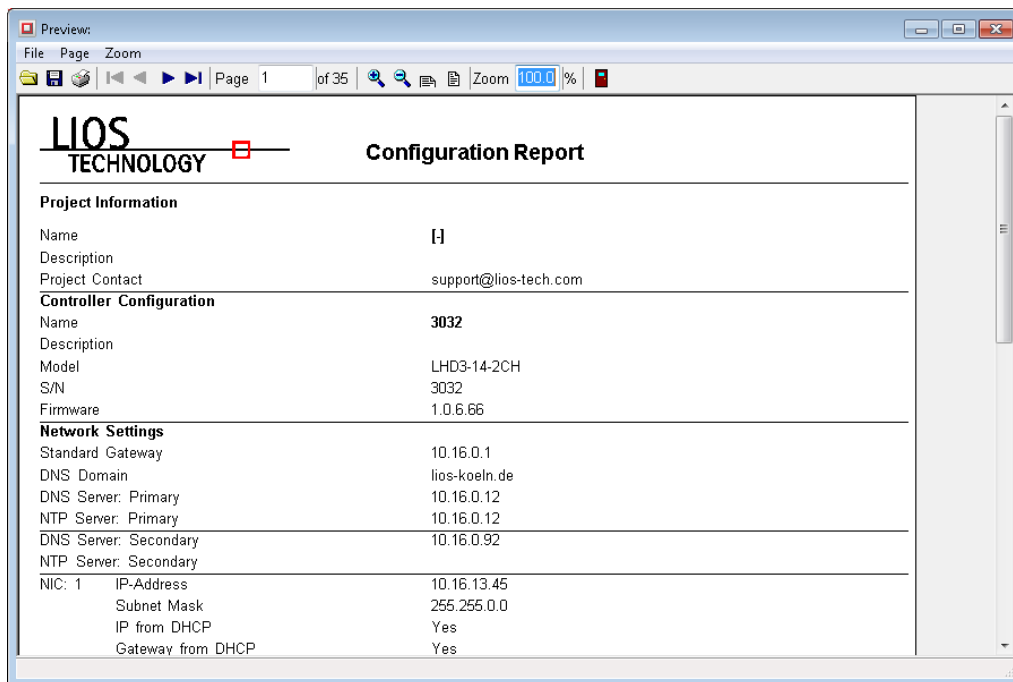


Figure 86: Configuration report



6.3.3 Status Report

A controller status report is generated automatically by pressing the STATUS REPORT button of the SERVICE tab (Figure 87). The report includes controller and fibre status information.

The controller status information comprises

- project information,
- the software and firmware revisions,
- the current internal temperatures,
- the current level of supply voltage,
- the total operating time,
- and the maximum and minimum levels of the internal temperatures and the supply voltage.

The fibre status information comprises

- the length of fibre, and
- the values of relevant optical attenuation parameters

for each measurement channel.

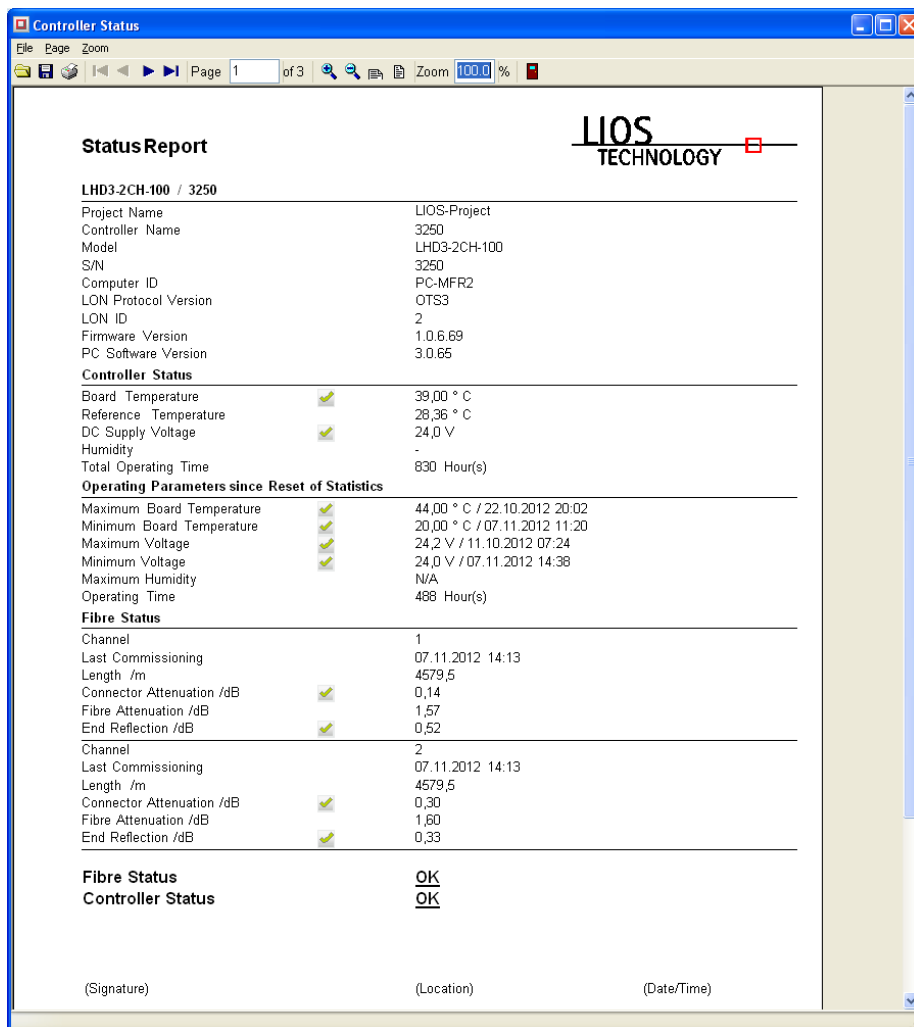


Figure 87: Status report



The report can be saved or printed using the respective buttons in the upper left corner of the CONTROLLER STATUS window.

6.3.4 Project Information

The controller may be assigned to a project, and the project information is saved in the controller memory. The project information is displayed in the PROJECT INFORMATION window (Figure 88) after pressing the PROJECT INFORMATION button of the SERVICE tab. You can edit project name, description and contact in the PROJECT INFORMATION window. Close the window by pressing OK to confirm the changes. Otherwise, press CANCEL.

Figure 88: Project information window

6.3.5 Support Information

Caution!

Perform at least one full measurement cycle on all fibres after the last change of configuration or power off and on the controller before generating the support information. Charon3 requests the support information from the controller. This requires that the controller is powered on and the Charon3 tree view indicates an active communication. Otherwise, the support information may be incomplete.

Support information of the controller is prepared after pressing the SUPPORT INFORMATION button of the SERVICE tab. First, the SUPPORT INFORMATION window opens (Figure 89). Then, diagnosis data is downloaded from the controller. Wait till the red message **DOWNLOADING DIAGNOSIS DATA. PLEASE WAIT.** (Figure 89) is replaced by the black message **DIAGNOSIS DATA DOWNLOADED** (Figure 90).

Enter the requested information into the respective fields of the GENERAL tab. Mandatory fields have a red headline that changes to black after entering the information (Figure 90).

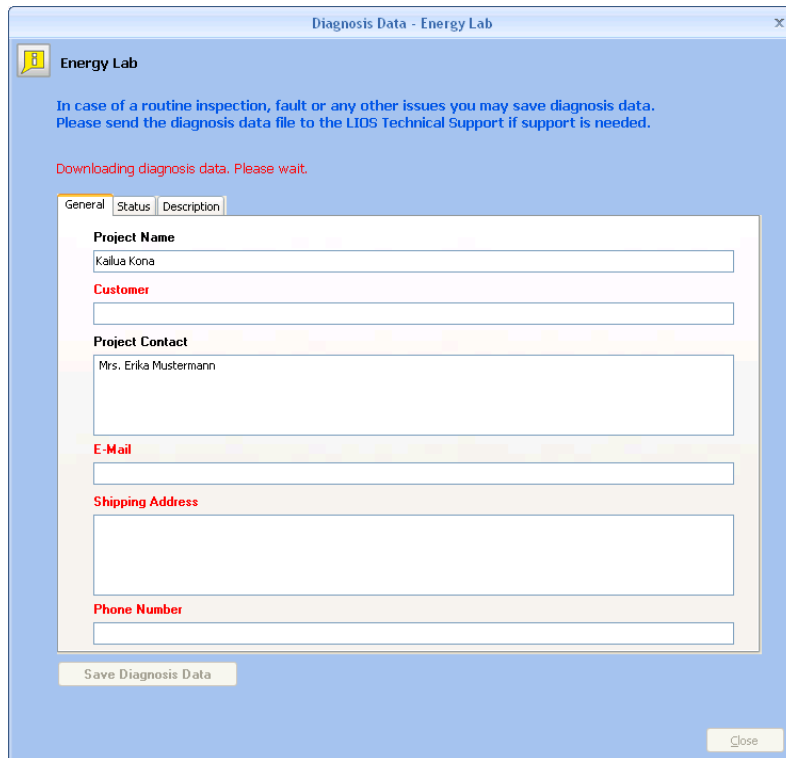


Figure 89: Support information window

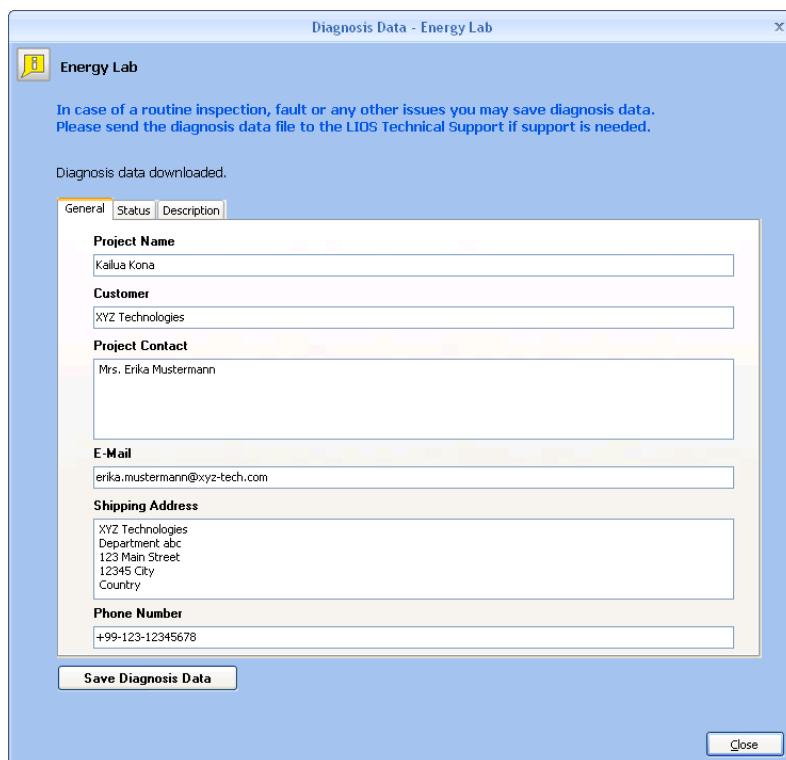


Figure 90: Diagnosis Data - General tab after entering the information

At the STATUS tab (Figure 91), check the applicable buttons and boxes for type of intervention, type of issue, temperature reading issues and environmental conditions.

In case of an emergency or fault, enter a short description of the issue, the countermeasures taken so far and their result at the DESCRIPTION tab (Figure 92). Describe faults as precisely as possible.



All support information is saved as a support file (*.spp) by pressing the SAVE DIAGNOSIS DATA button. We recommend saving of support information at least once a year.

In case of an emergency of fault, send the support information to the LIOS Support by e mail (Charon3@support.lios-tech.com).

Close the window by pressing OK.

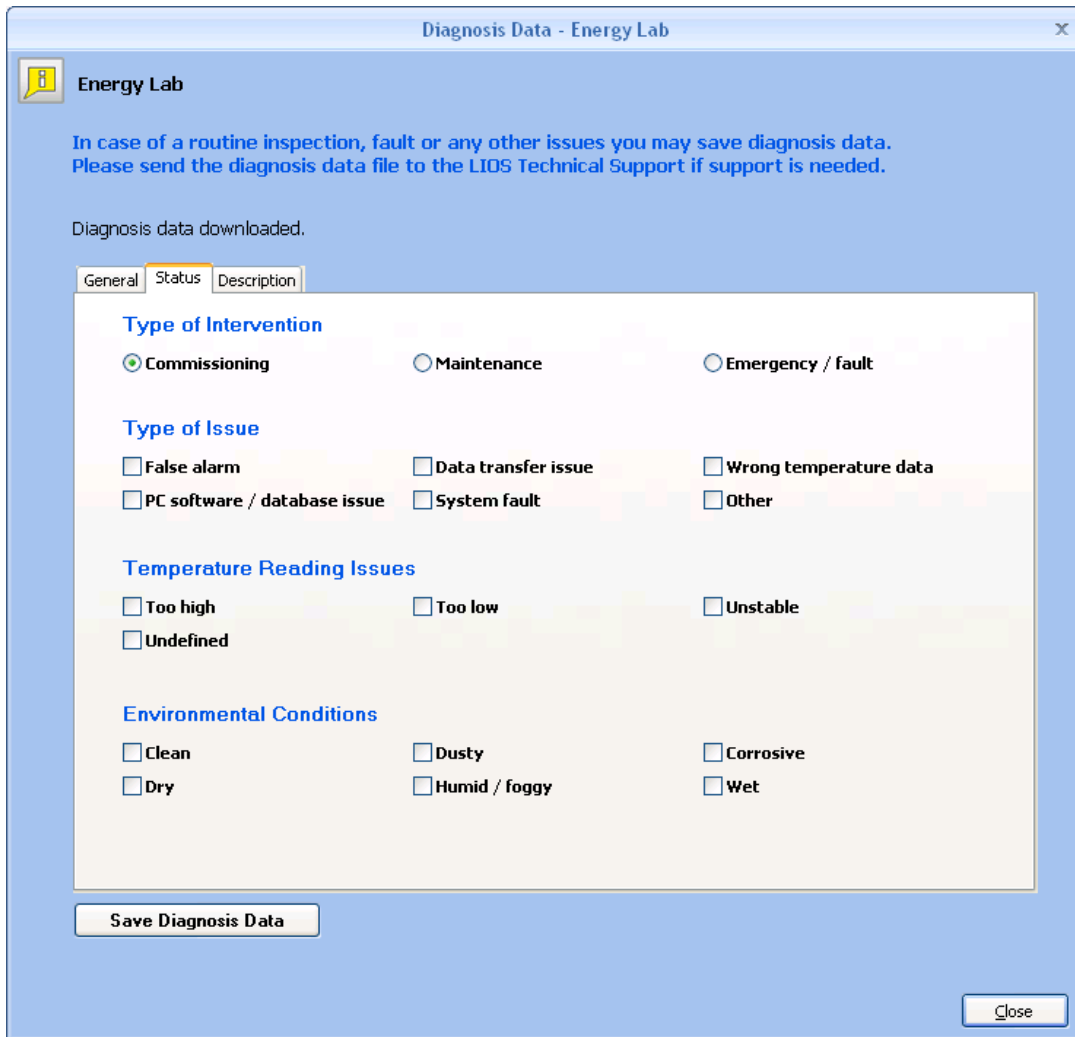


Figure 91: Diagnosis Data - Status tab

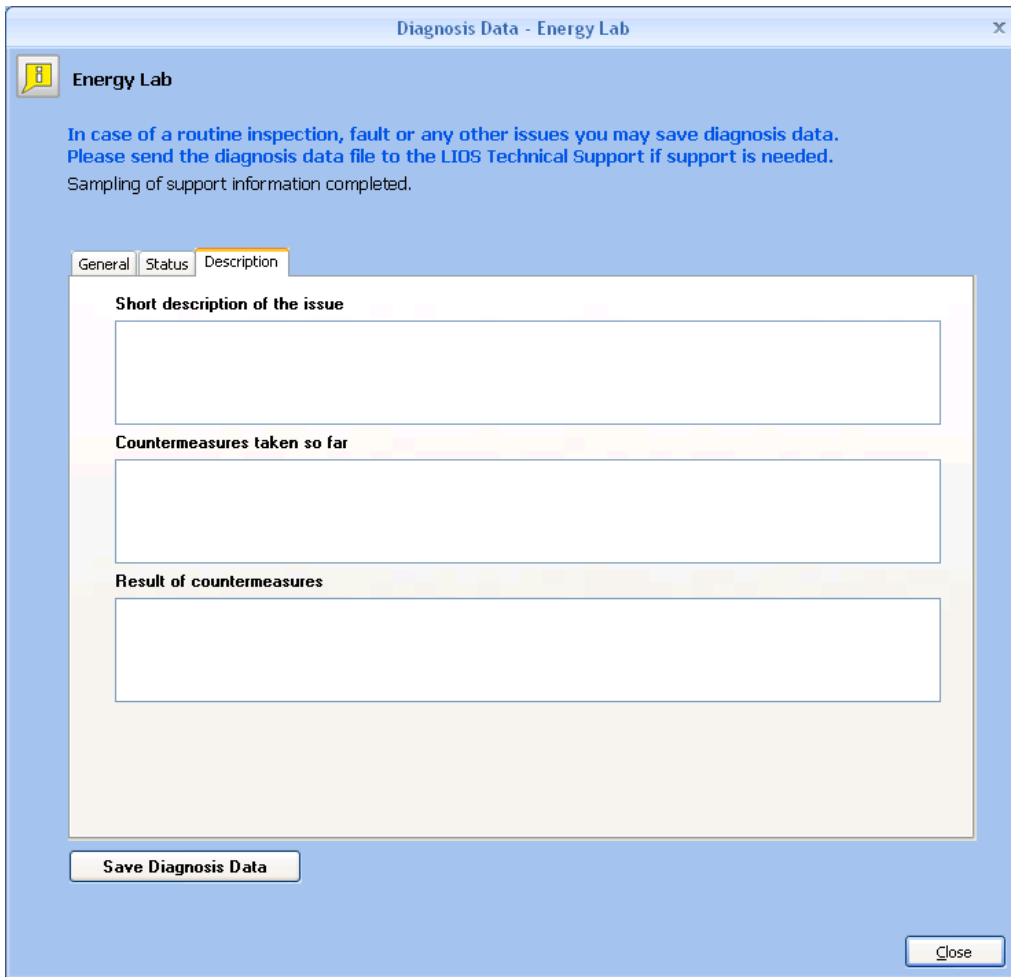


Figure 92: Diagnosis Data - Description tab

6.3.6 Firmware Update

The firmware of the controller can be updated by pressing the FIRMWARE UPDATE button of the SERVICE tab. Then, the SELECT FIRMWARE window (Figure 93) opens. Please choose the proper location and filename of the firmware to be used and press OPEN.

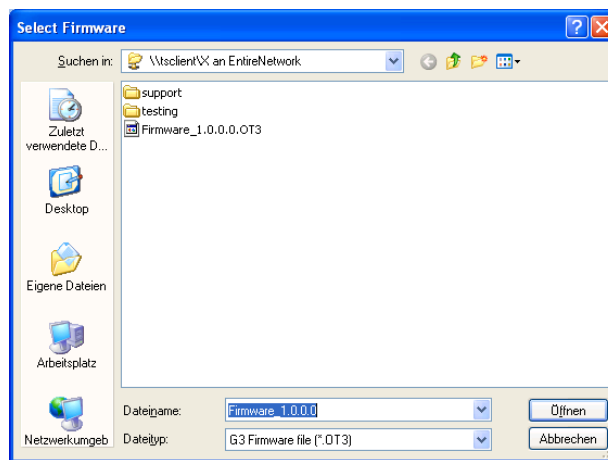


Figure 93: Select firmware window



Then, the FIRMWARE UPLOAD window (Figure 94) opens. You need to agree to the firmware license to start the upload. Then, the FIRMWARE UPLOAD CONFIRMATION window (Figure 95) opens. Press OK to start the upload.

The upload takes about 2 minutes. The progress of firmware upload and erase is shown in the status section of the TREE VIEW (Figure 96). The controller re-starts automatically after successful upload.

Caution!

Never switch off the controller or disconnect power during the upload or during erasing of old firmware! Wait until the re-start has taken place! Power loss during firmware upload may make a factory repair at LIOS Technology necessary.

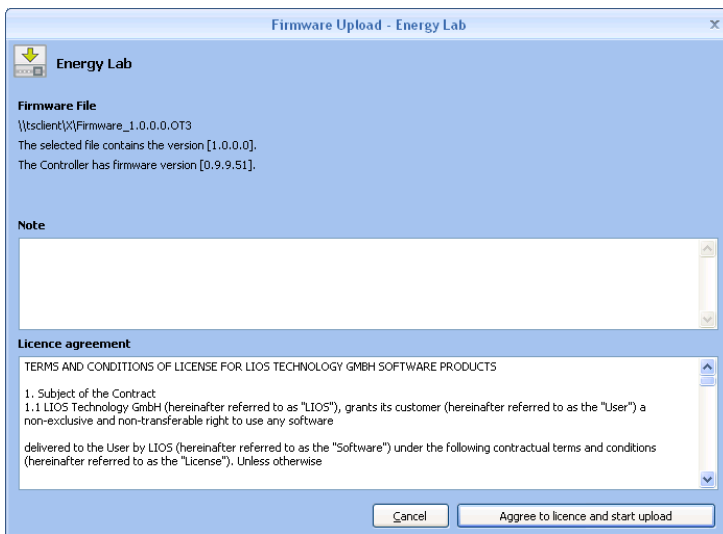


Figure 94: Firmware license agreement



Figure 95: Firmware upload confirmation window

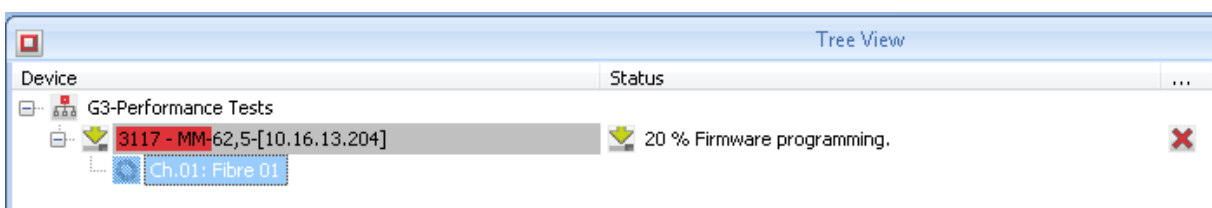


Figure 96: Tree view during firmware upload



6.4 Optional Licenses

Software functions of the controller and Charon3 can be extended using optional licenses. Table 9 shows an overview of the available licenses. You may purchase the licenses from LIOS Technology and load the license file using the import function of Charon3 (see section 8.1).

Table 9: Optional licenses

Name of License	Description
Sampling interval 0.25 m ⁵	Minimum sampling interval of 0.25 m
Temperature data over LON	Output of temperature and backscattering profiles over the LON protocol
Controller fieldbus data server	Output of controller data over the MODBUS TCP, DNP3 and IEC 60870 protocols
BACNET ⁶	Output of controller data over the BACNET protocol
Charon3 Visualization	Connection of the controller to Charon3 for unlimited time
Charon3 Enhanced View	Enhanced visualization features of Charon3

7 Data Visualization

7.1 Quick Access Toolbar

The QUICK ACCESS TOOLBAR is shown in the upper left corner of the Charon3 window (Figure 97). It offers buttons for printing, opening the measurement explorer, importing data and customizing the toolbars.

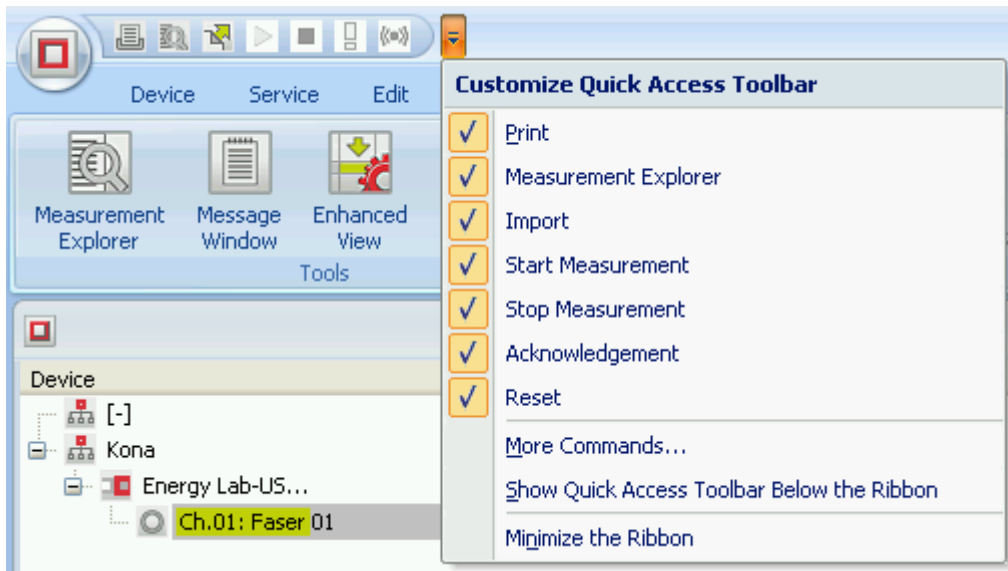


Figure 97: Quick access toolbar

The CUSTOMIZE window (Figure 98) opens after clicking to MORE COMMANDS. It offers options for a display of the different toolbars by checking or unchecking the respective boxes.

⁵ Distance range limited to 1 km.

⁶ Available on request.

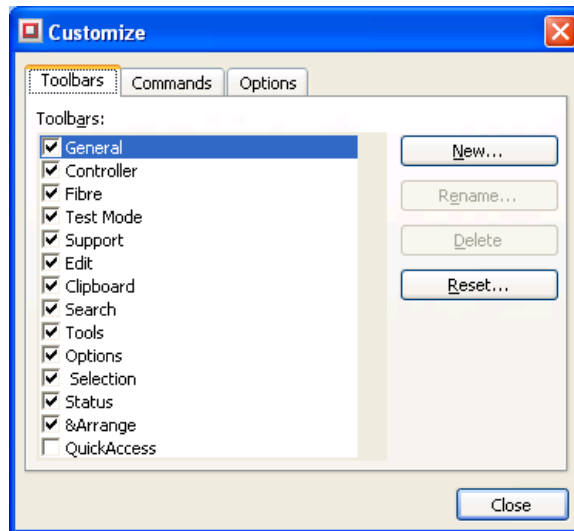


Figure 98: Toolbar tab of the customize window

The OPTIONS tab (Figure 99) of the CUSTOMIZE window offers further options for the appearance of the Charon3 window.

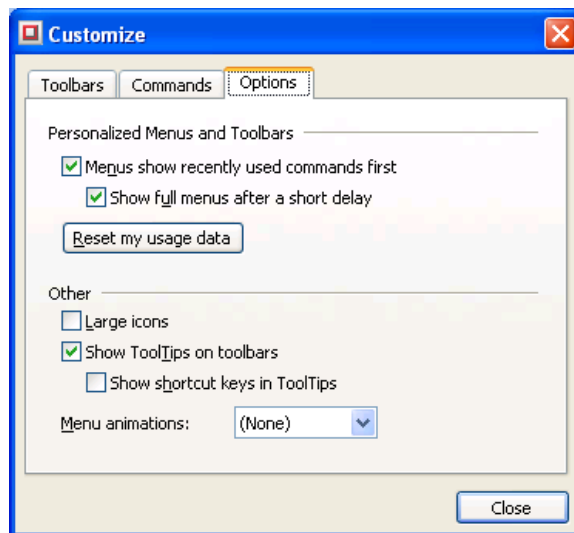


Figure 99: Options tab of the customize window



7.2 Edit

The name of a controller or fibre activated in the TREE VIEW can be edited using the RENAME button of the EDIT tab. Activated controllers can be deleted from the tree view using the DELETE button of the same tab.

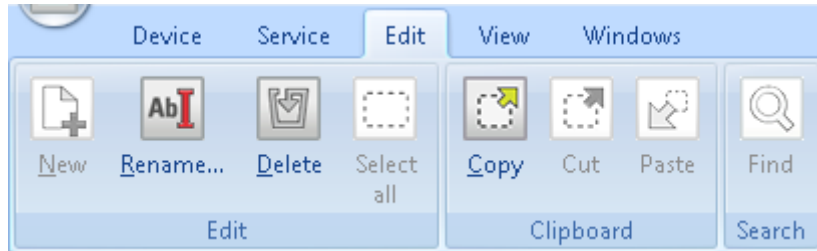


Figure 100: Edit tab

7.3 View

The displayed data and program options are handled using the commands shown in the VIEW tab (Figure 101). The buttons of this tab are grouped into TOOLS and OPTIONS.



Figure 101: View tab

7.3.1 Tools

The group TOOLS of the VIEW tab comprises the buttons MEASUREMENT EXPLORER, MESSAGE WINDOW, HISTORY and REFRESH.

7.3.1.1 Measurement Explorer

The measurement explorer is a powerful tool for managing the Charon3 database and for selecting data for visualization and export. The MEASUREMENT EXPLORER window (Figure 102) opens after pressing the MEASUREMENT EXPLORER button in the TOOLS section of the VIEW tab (Figure 101).

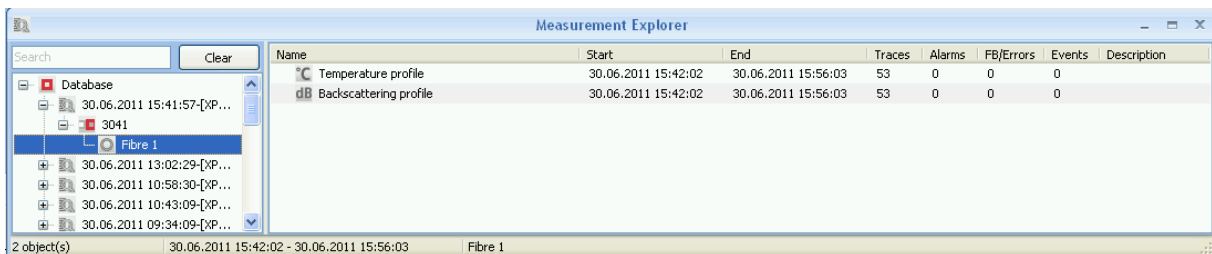


Figure 102: Measurement explorer window



The left part of the MEASUREMENT EXPLORER window shows the content of the database. The database content is organised by measurement sessions labelled with date, time and user name. Each session can contain temperature and backscattering profiles from a number of controllers and fibres as well as zone data. The active measurement session is marked by using a bold font. Single controllers or fibres can be selected in the left part of the window.

Single or multiple TEMPERATURE PROFILE (Figure 120) and BACKSCATTERING PROFILE (Figure 122) windows can be opened by double clicking controller, fibre, temperature profile or backscattering profile entries in the right part of the MEASUREMENT EXPLORER window.

Double clicking on a measurement session entry opens the zone view (Figure 123), temperature profile and backscattering profile windows for all data sets in the session.

Measurement sessions can be completely or partially deleted using the pull down menu appearing at a right mouse click on the measurement session entry (Figure 127). The PROPERTIES entry of that pull down menu can be used to add a description to the measurement session.

7.3.1.2 Message Window

The MESSAGE WINDOW opens after pressing the MESSAGE WINDOW button in the VIEW tab. It shows the detailed communication between the controllers and Charon3. The MESSAGE WINDOW is not needed for normal operation. It can provide additional information in case of communication issues.

Time	Source	Message
19.04.2011 15:59:14	3041-HP:[10.16.13.50]	DSP: Fiber 1 - OFDR data acquisition progress 35%
19.04.2011 15:59:09	3041-HP:[10.16.13.50]	DSP: Fiber 1 - OFDR data acquisition progress 10%
19.04.2011 15:59:09	3032-HP:[10.16.13.51]	close connection 1 to pc-mwi3.lios-koeln.de
19.04.2011 15:59:09	3032-HP:[10.16.13.51]	read from peer 1 failed (0): (null)
19.04.2011 15:59:04	TSubDataZoneMin	Zone Minimum" [19,21 deg C@6,00 m]" [20,07 deg C@201,00 m]" [19,93 deg C@228,00 m]" [20,02 deg C@312,00 m]" [19,97 deg C@
19.04.2011 15:59:04	TSubDataZoneMax	Zone Maximum" [20,56 deg C@39,00 m]" [20,52 deg C@168,00 m]" [20,44 deg C@240,00 m]" [20,39 deg C@333,00 m]" [20,52 deg C
19.04.2011 15:59:04	TSubDataZoneMean	Zone Average:" 20,25 deg C" 20,30 deg C" 20,22 deg C" 20,19 deg C" 20,19 deg C" 20,11 deg C" 20,11 deg C" 20,08 deg C" ;
19.04.2011 15:59:04	TSubDataProfileRefTemp	Temperature data received.

Figure 103: Message window

7.3.1.3 Temperature History

The TEMPERATURE HISTORY window visualizes the temporal development of the temperature at certain locations. Just double click onto a data point in a TEMPERATURE window. Then, the TEMPERATURE HISTORY window (Figure 104) opens below the TEMPERATURE window and shows temporal development of the temperature at the selected location. The temporal development of the temperature at other locations can be added to the TEMPERATURE HISTORY window by double clicking the respective data points. In addition, the temporal development of the temperature at the mouse cursor position is shown in red. The legend in the lower part of the windows shows which locations correspond to the displayed temperature history curves.

Double clicking on a certain point in the temperature history curve automatically moves the scroll bar of the related temperature and backscattering profile windows to the temperature trace taken at that time.

The TEMPERATURE HISTORY window can also be opened by pressing the HISTORY button of the VIEW tab (Figure 101) while a TEMPERATURE window is activated.

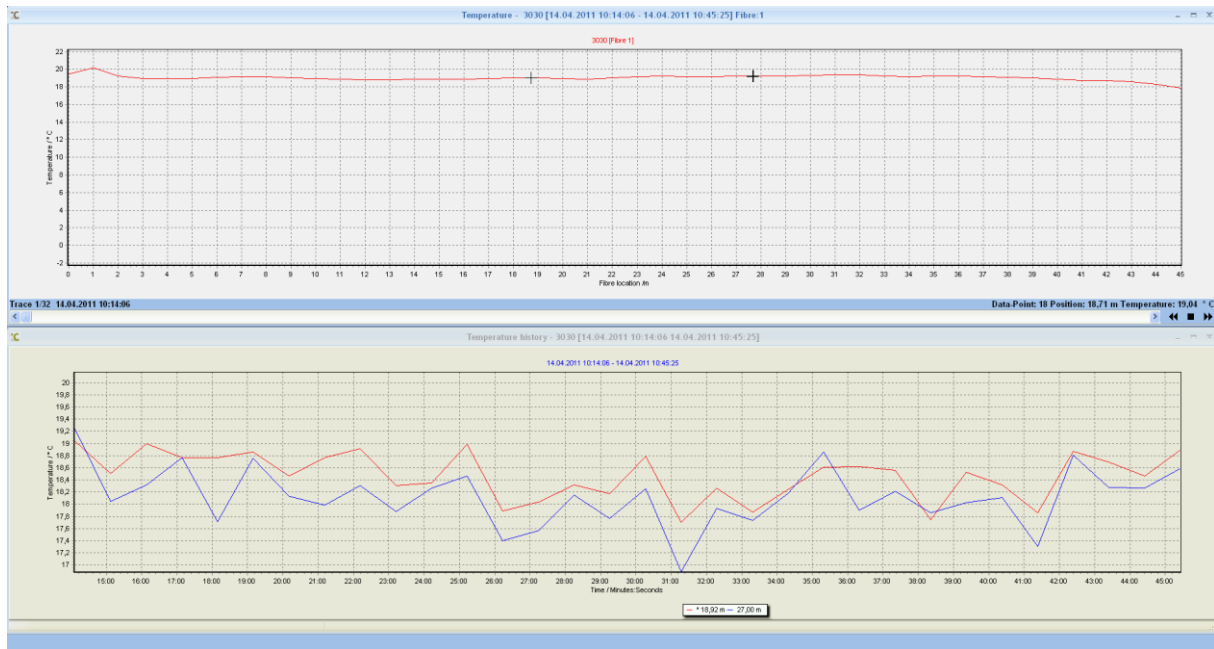


Figure 104: Temperature profile and temperature history windows

7.3.1.4 Refresh

Pressing the REFRESH button of the VIEW tab or the REFRESH entry of the pull down menu shown in Figure 127 updates the measurement session list of the activated measurement explorer window. The REFRESH may be used to get access to a recently taken measurement session.

7.3.2 Options

7.3.2.1 Options

The OPTIONS window (Figure 105) opens after pressing the OPTIONS button in the VIEW tab. It comprises the tabs GENERAL, DEFAULT CONFIGURATION, CLEANUP, MEASUREMENT SESSIONS and EXPORT.

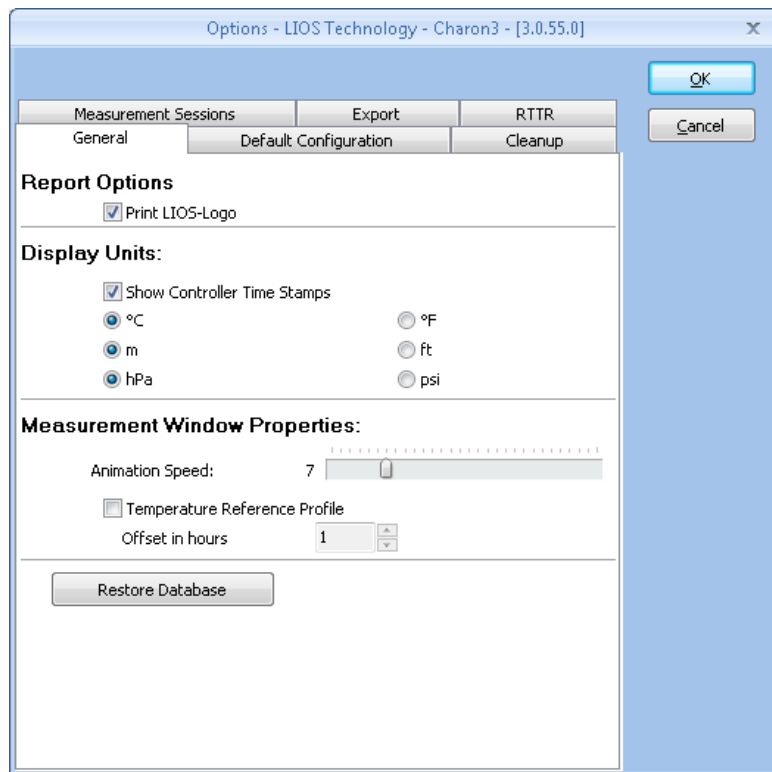


Figure 105: General tab of the options window

In the GENERAL tab (Figure 105) you can choose the following options:

- The LIOS logo is printed in the reports if the corresponding checkbox is activated.
- You can select CONTROLLER TIME STAMPS as well as international (SI) or US units for the display of data. Controller time stamps correspond to Universal Time (UT), whereas PC time stamps usually correspond to the local time. Charon3 needs to be restarted after changing the units. Data in the database are always saved in SI units. US units can only be used for the display and for the ASCII export of data.
- You can adjust the ANIMATION SPEED of the measurement window according to the capability of your PC and your personal requirements.
- You can choose to display a REFERENCE TEMPERATURE PROFILE with a selectable time offset to the current temperature profile. You can specify the time difference between the current and the reference temperature profiles by setting the OFFSET IN HOURS.
- In case of a fault in the Charon3 database, you can restore the configuration data of the controllers in the tree view by pressing the button RESTORE DATABASE.

Caution!

All measurement data in the database are deleted by restoring the database.

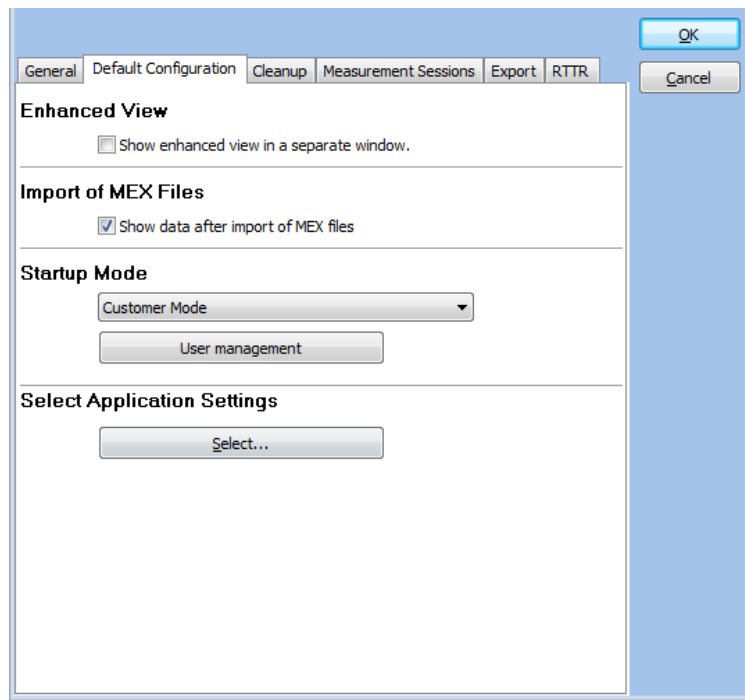


Figure 106: Default configuration tab of the options window

In the DEFAULT CONFIGURATION tab (Figure 106) you can choose the following options:

- You can choose to display an ENHANCED VIEW in a separate window by checking the corresponding box. (ENHANCED VIEW is an optional feature of Charon3 requiring a special license.)
- You can choose to open the measurement data windows automatically after a *.mex file import.
- You can select two different modes for starting Charon3:
 - o Customer mode (default) – Charon3 starts in user level 1
 - o Service Mode – Charon3 starts in user level 3. This mode can only be activated in user level 3 (see section 3.3).
- You can select different application settings.

The APPLICATION SETTINGS cover the following:

- Authorisation for changing or selecting of controller settings (general settings, standard zones, measuring and alarm parameter sets)
- Default settings for the initial commissioning.

The APPLICATION SETTINGS were assigned to a controller during the initial commissioning within the QUICKSTART. In case of an already commissioned controller, it has to be reset to factory defaults, before APPLICATION SETTINGS can be reassigned. This procedure avoids the unwanted change of controller settings in case of subsequent modifications of the APPLICATION SETTINGS.

Press the select button to select application settings. The APPLICATION SETTINGS window opens (Figure 107). It shows a list of pre-defined application settings. You can activate an application setting by clicking the respective ACTIVE button. You can also open new application settings or show and delete existing ones.



In user level 4 (see section 3.3), you also can copy, edit or save application settings (Figure 108). Use the combination of copy and edit to generate a new application setting. Name, description and author of an application setting can be edited in the list by clicking the respective field and pressing the F2 function key.

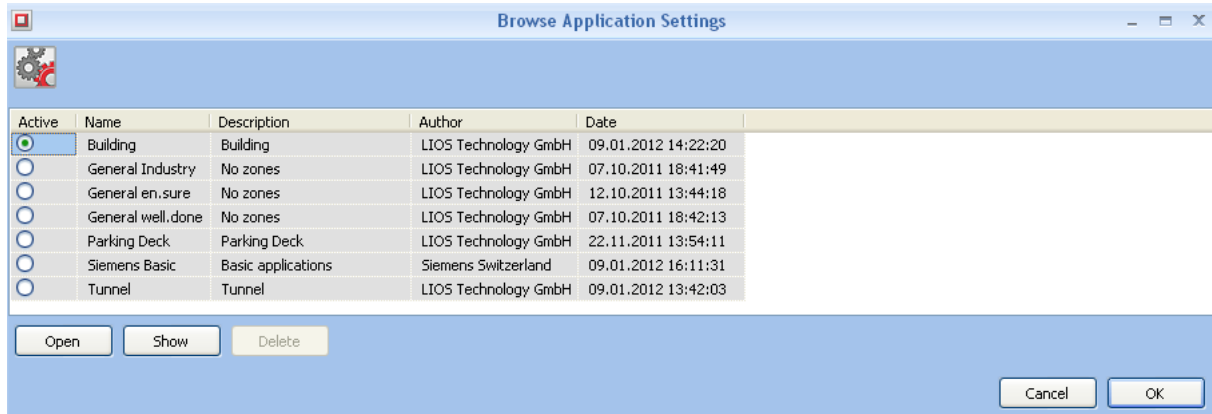


Figure 107: Application settings window

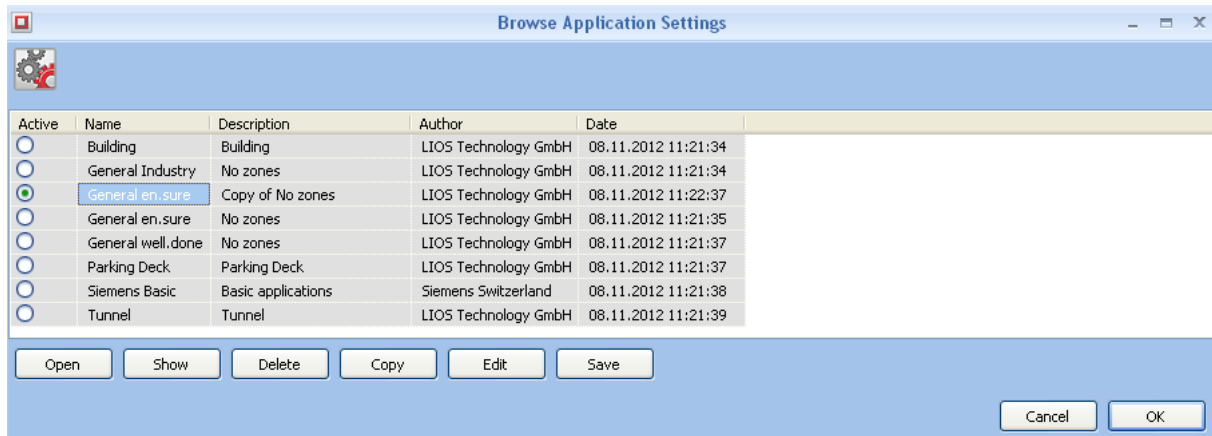


Figure 108: Applications settings window in user level 4

Select an application setting by clicking its name and using the SHOW or EDIT buttons to open it. Then, the APPLICATION SETTING window (Figure 109) opens. It shows a list of Charon3, controller, test mode and parameter set settings.

In the edit mode of user level 4, you can choose the default settings by selecting options or by editing the entries in the DEFAULT SETTINGS column. You can also set permissions, meaning that a user cannot change the respective setting when the application setting is active.

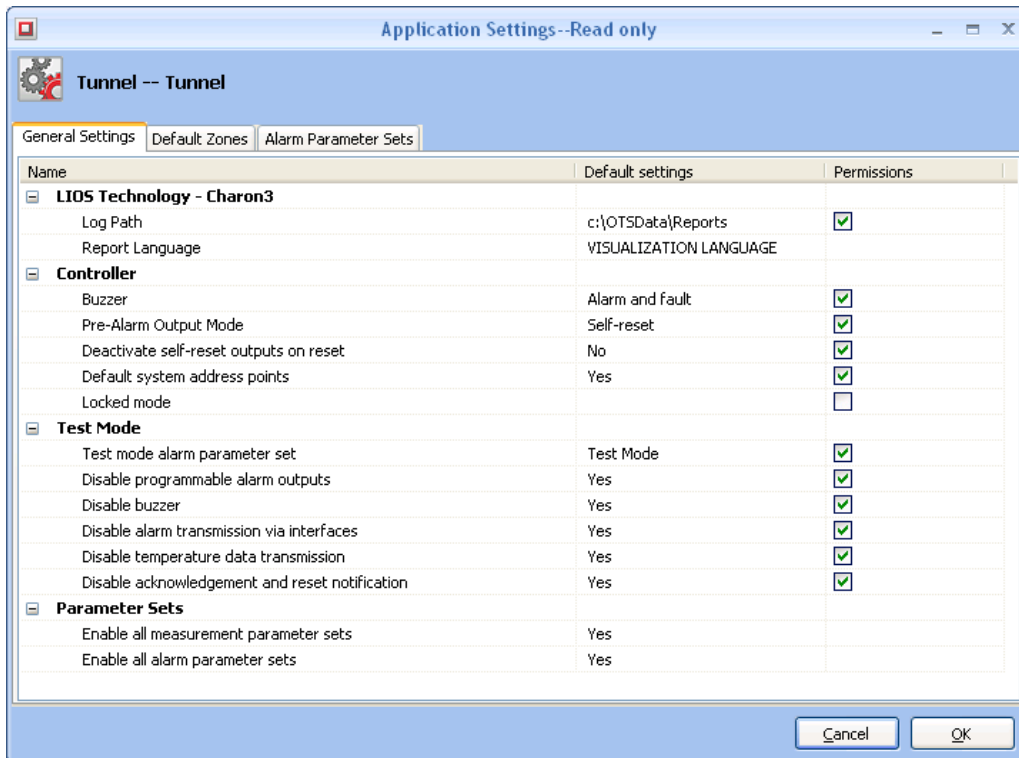


Figure 109: Application settings window - general tab

The DEFAULT ZONES tab shows a table of feed, alarm and runout zone settings. Those zones are automatically defined, when the application setting is selected during the QUICK START (section 4.1). The software defines one feed zone, one runout zone and a number of alarm zones with the pre-defined zone lengths and alarm parameter sets together covering the whole length of the sensor cables. The names of the alarm zones are generated from the name in the table and a serial number.

In the edit mode of user level 4, you can change the default zone settings in the table using the double click or the function key F2.

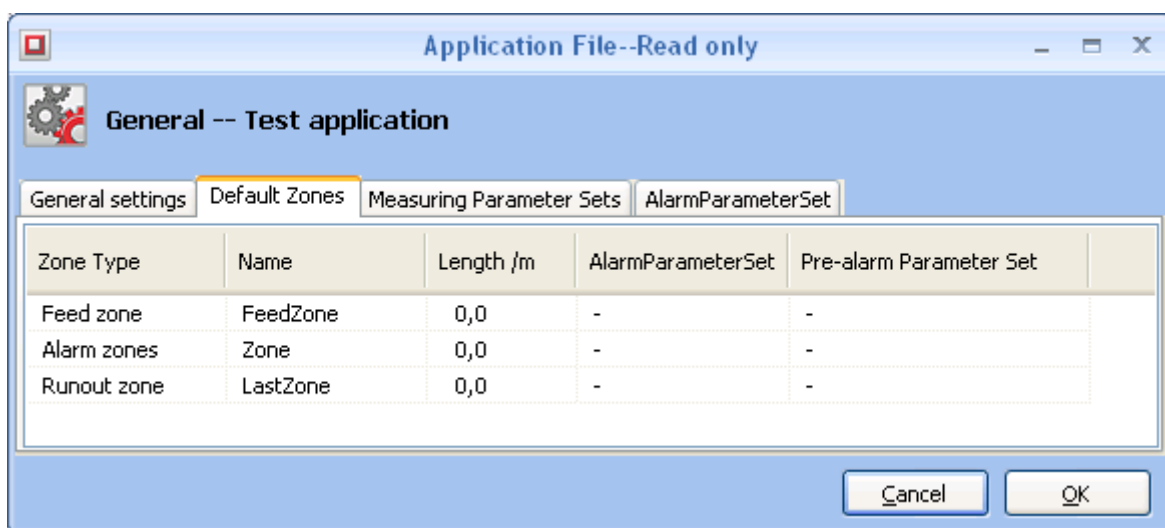


Figure 110: Application settings window - default zones tab

If the ENABLE ALL MEASURING PARAMETER SETS option at the general tab is set to NO, the activation of measuring parameter sets is shown at the MEASURING PARAMETER SETS tab (Figure 111).



Only measuring parameter sets with a checked PERMISSION box are available, when the application setting is activated.

In the edit mode of user level 4, you can change the permission settings by checking or un-checking the boxes.

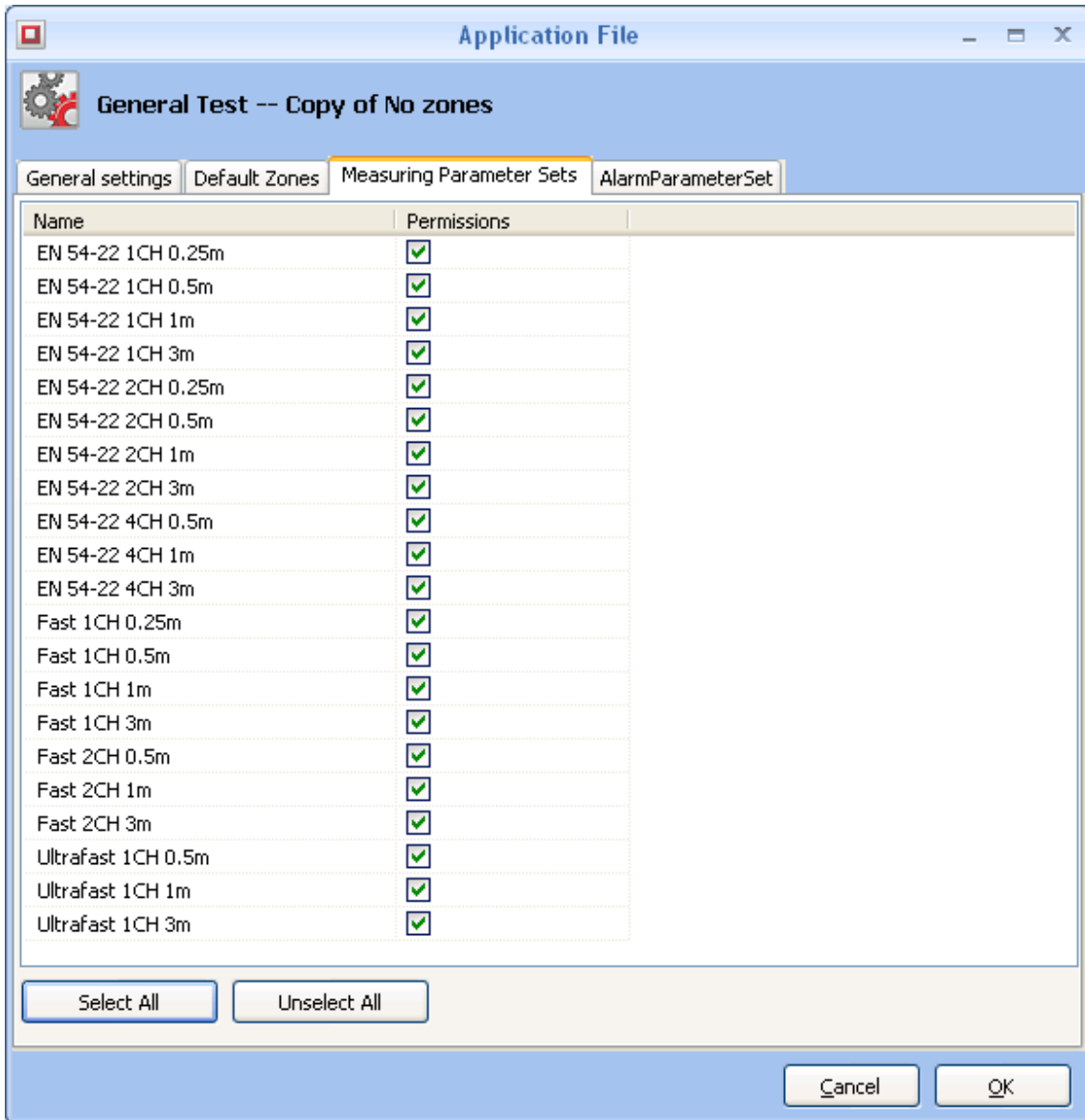


Figure 111: Application settings window - measuring parameter sets tab

The activation of alarm parameter sets is shown at the ALARM PARAMETER SETS tab (Figure 112). If the ENABLE ALL ALARM PARAMETER SETS option at the general tab is set to NO, the check boxes represent permissions. Otherwise, they just indicate which alarm parameter sets are included in the application setting when it is saved to a file. Only alarm parameter sets with a checked PERMISSION box are available, when the application setting is activated.

In the edit mode of user level 4, you can change the permission settings by checking or un-checking the boxes.

Press OK to close the application setting or to save the changes.

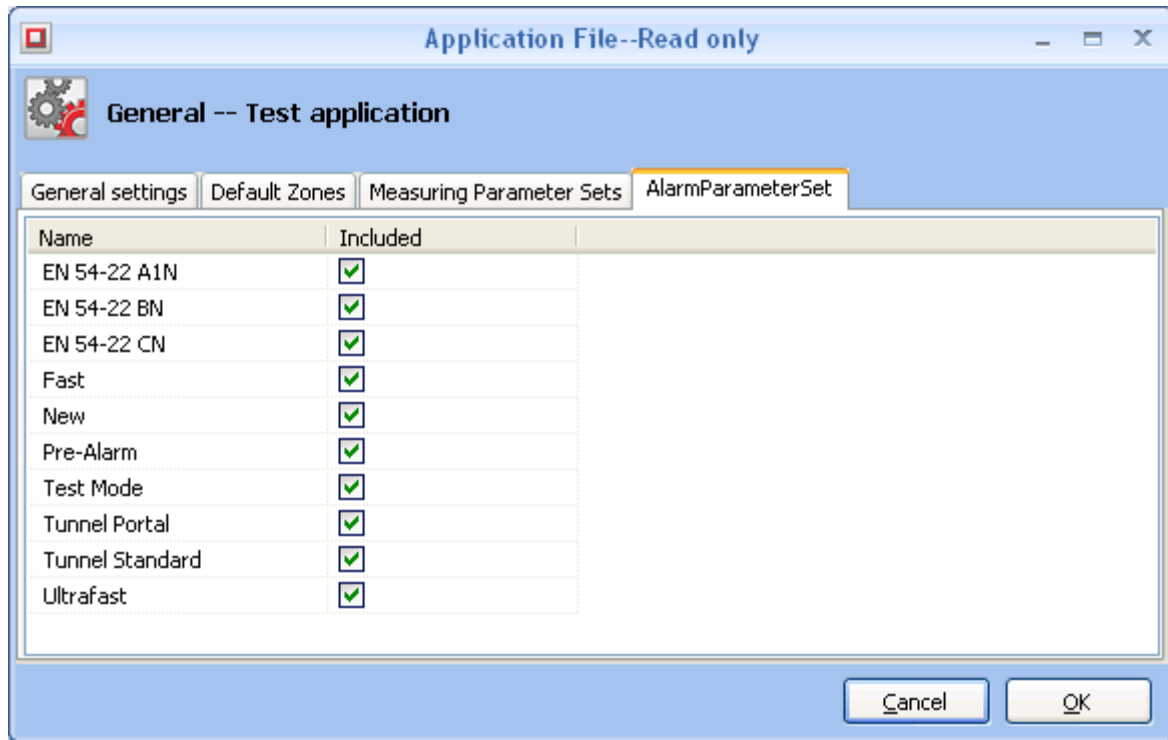


Figure 112: Application settings window - alarm parameter sets tab

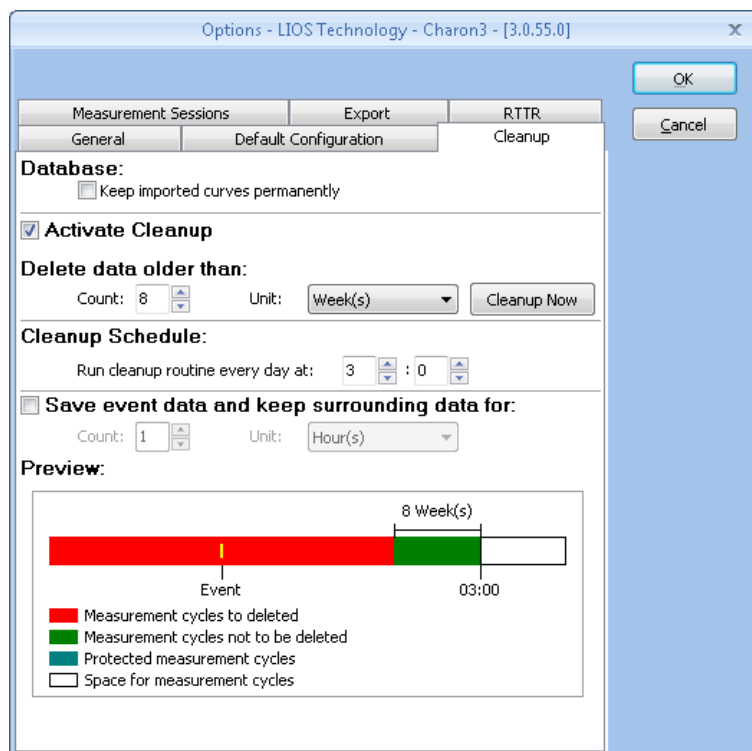


Figure 113: Cleanup tab of the options window

Charon3 saves all measurement and configuration data in a database. Considerable volumes of data can be accumulated over time, although all data is stored in a compressed format. It is therefore essential to choose proper settings for a database cleanup in the CLEANUP tab (Figure 113).

- You can select to keep imported data permanently. Then, this data is excluded from cleanup.



- You can activate the automatic cleanup and select a suitable age of data to be deleted. You can also start an immediate cleanup by pressing CLEANUP NOW.

Caution!

Older data cannot be restored after cleanup. Export any permanently required measurement data to a *.mex or ASCII data file before starting a cleanup.

Please estimate the volume of data to be accumulated over time, compare it to the available capacity of your hard disk, and choose the cleanup settings accordingly. Otherwise, the database can get faulty, and all data can be lost. You should also keep the volume of the database as low as possible in order to achieve optimum performance of the Charon3 visualization.

- The automatic cleanup is executed during each shutdown of Charon3. You can also schedule a daily cleanup for a time where optimum performance of the visualization is not required.
- You can choose to keep older event data and measurement data surrounding events in the database after cleanup. This prevents the loss of important data by the cleanup.

The preview in the bottom of the cleanup tab gives you an estimate on the portion of data to be deleted in the next cleanup. The cleanup does not immediately reduce the size of the database. Every 30 days and in case of low free disk space, Charon3 automatically initialises a compression of the database that releases the disk space of the deleted entries. Depending on the volume of the database, the compression can take several minutes.

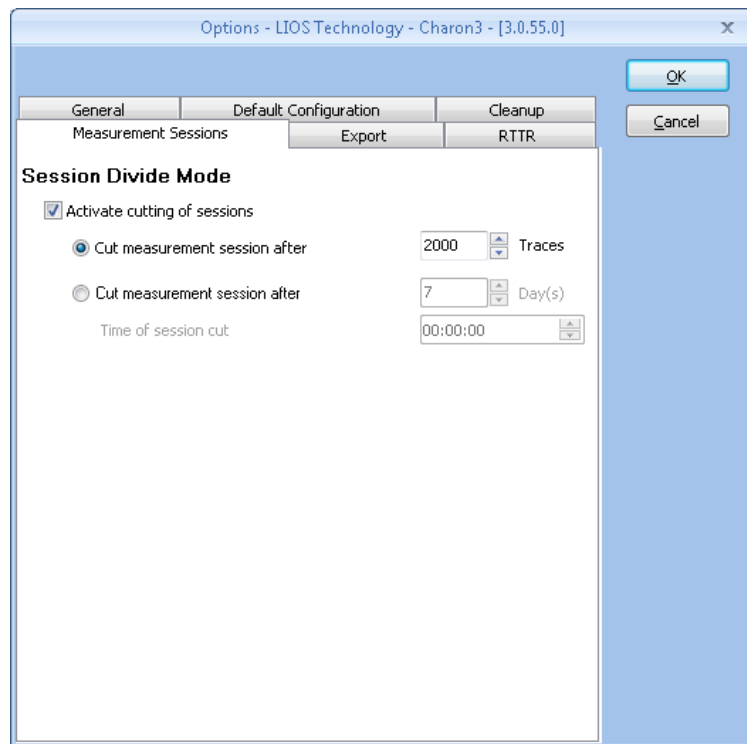


Figure 114: Measurement sessions tab of the options window

Charon3 organises the measurement data in the database in a number of sessions in order to enhance the performance of the data visualization. You can change the measurement session settings in the MEASUREMENT SESSIONS tab (Figure 114) of the options window.



We strongly recommend activating the cutting of sessions. You can choose to cut sessions after a number of temperature profiles (traces) or after a number of days. In the latter case, you can also set a time for cutting the sessions. The optimum number of traces in a session depends on the capability of your PC. We recommend using a maximum of 2000 traces in the beginning.

You can activate the automatic export of temperature profiles by checking the box on the EXPORT tab of the OPTIONS window (Figure 115). You can also choose the path for saving the exported data. If the export is activated, Charon3 automatically saves new data received from connected controllers with a ONLINE DATA EXPORT to the chosen folder.

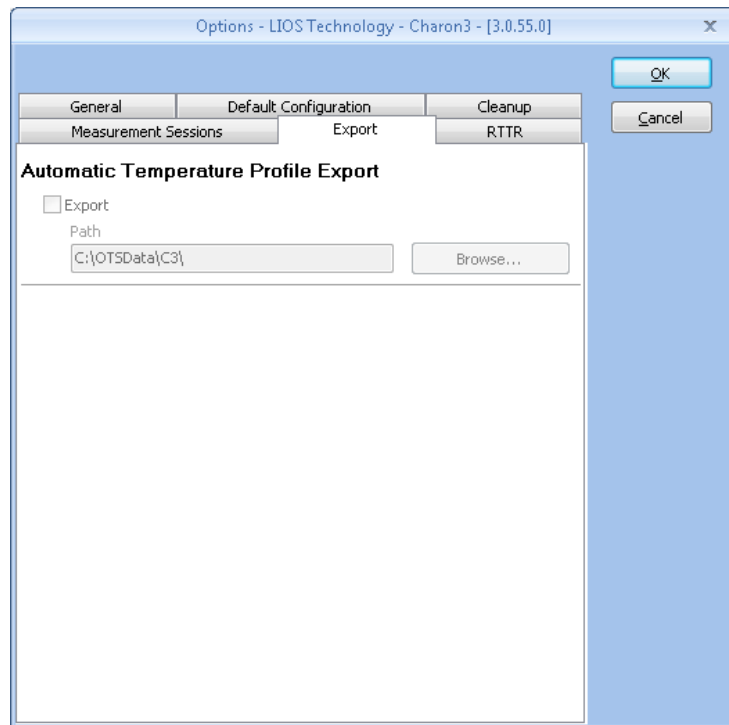


Figure 115: Export tab of the options window



7.3.2.2 Language

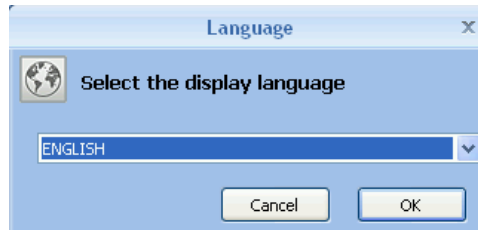


Figure 116: Language window

Charon3 supports a number of different languages. The LANGUAGE window (Figure 116) opens after pressing the LANGUAGE button in the OPTION tab. Select your language in the pull down menu and press OK.

7.3.2.3 Licence

(Not yet implemented. The LICENCE button will enable the installation of additional software licences in future versions of Charon3.)

7.3.2.4 Plug-In

The PLUG-IN button enables the activation and configuration of additional software features. In Figure 117 one plug-in is shown. Additional plug-ins are available for the implementation of communication protocols, data export function and continuative data processing.

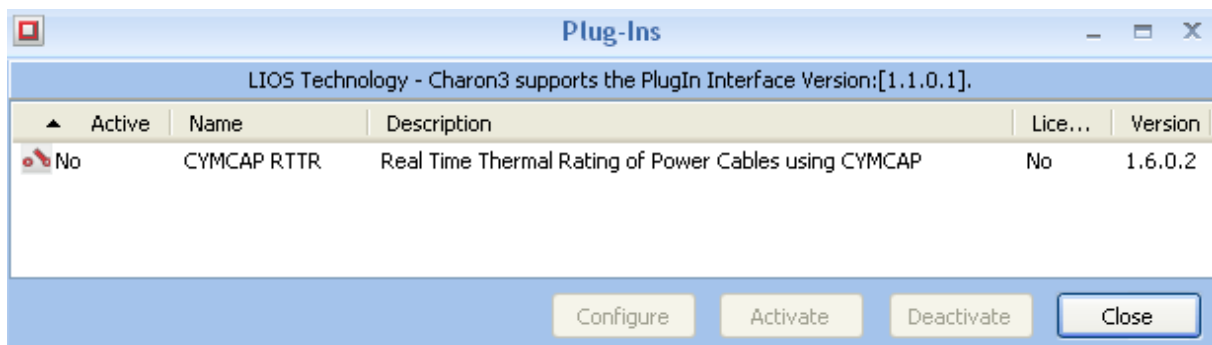


Figure 117: Plug-In Window

7.4 Windows

The displayed windows are handled using the commands shown in the WINDOWS tab (Figure 118). You can select the active window using one of the buttons of the left section SELECTION. For instance, if you press the SHOW TEMPERATURE DATA button, a list of open temperature data windows is displayed, and you can activate one of them by clicking its entry (Figure 119). Further types of data windows can be activated using the buttons SHOW BACKSCATTER DATA, SHOW ZONE DATA or WINDOWS BY NAME.

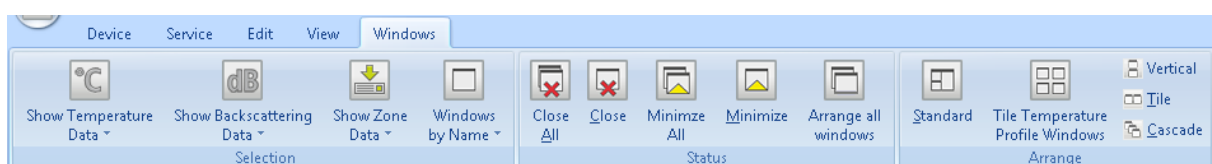


Figure 118: Windows tab

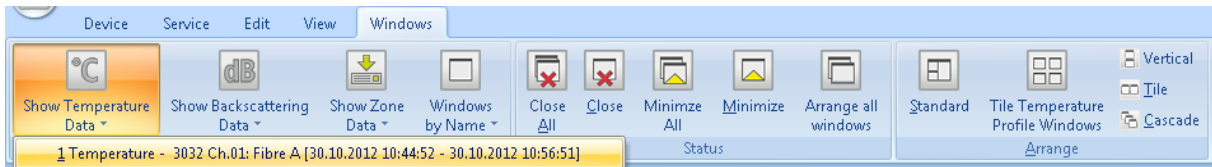


Figure 119: Activation of a temperature data window

All or individual windows can be closed or minimized using the buttons in the middle section STATE. The arrangement of windows on the screen can be changed using the buttons in the right section ARRANGE.

7.4.1 Temperature Profile

The TEMPERATURE PROFILE window (Figure 120) shows temperature profiles taken in a measuring sequence. You can move through the measurement sequence using the scroll bar at the bottom of the window. You can start and stop an automatic scrolling using the buttons at the lower right corner of the TEMPERATURE PROFILE window.

Number of data point, location and temperature reading at the cursor position are displayed in the lower right corner of the window.

If you click to a position in the temperature profile and move the mouse cursor to another position, a bold cross is displayed at the click position and a thin cross is displayed at the current mouse cursor position.

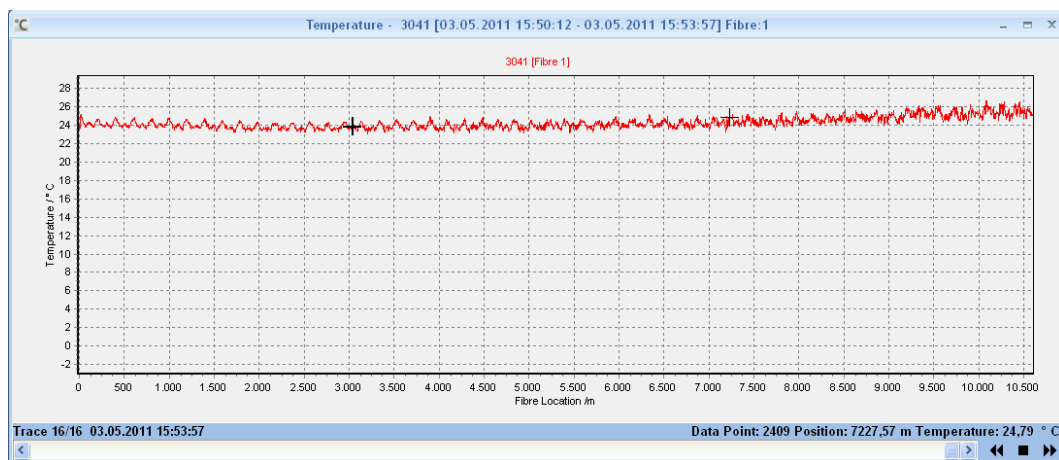


Figure 120: Temperature profile window



You can zoom into the temperature profile by pressing the left mouse button on the upper left corner of the desired section and by dragging the mouse pointer down to the lower right corner while the mouse button is permanently pressed. Then, release the mouse button. You can zoom out by pressing the mouse button at any position, drawing to the upper left and releasing the button.

You can scroll through the temperature traces using the button or the arrow keys of the scroll bar. Temperature traces of all channels are shown in the order of recording. If you click to the left or right side of the scroll bar, the next temperature trace of the same channel is shown. You can start and stop automatic scrolling of traces using the double arrow and square buttons at the lower right corner of the window.

You can choose any temperature trace to become permanently visible in the window by using a right mouse click and selecting SAVE CURVE TO REPOSITORY in the pull down menu (Figure 121). The traces in the repository and the trace selected by the scroll bar are then simultaneously visible in the window.

The scrolling and curve repository functions are also available for the backscattering profile windows (see section 7.4.2).

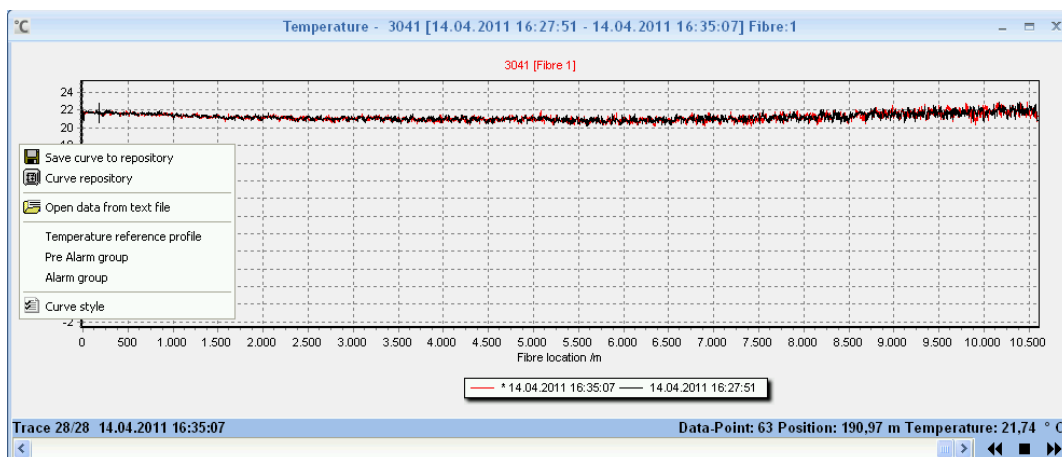


Figure 121: Curve repository in the temperature profile window



7.4.2 Backscattering Profile

The BACKSCATTERING PROFILE window (Figure 122) shows the Stokes, anti-Stokes or both backscattering profiles in linear or logarithmic presentation according to the selections made in the right part of the window. You can scroll as well as zoom in and out in the same way as described for the TEMPERATURE PROFILE window (section 7.4.1). The backscattering profiles are useful to evaluate the quality of the sensor cable installation (see section 5.4). For the description of scrolling and curve repository functions see section 7.4.1.

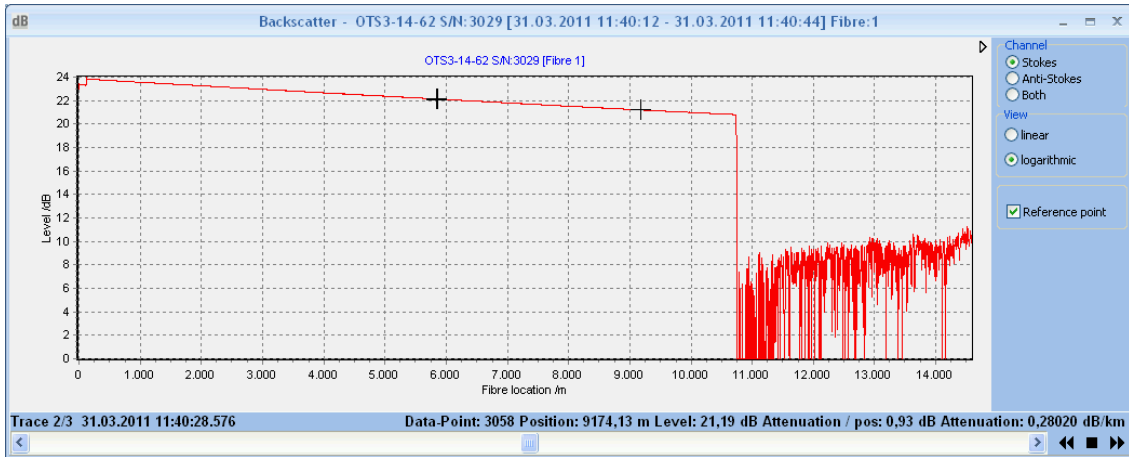


Figure 122: Backscattering profile window

7.4.3 Zone Data

The ZONE DATA window (Figure 123) opens after taking the first measurement in a measurement sequence or after double clicking on a measurement session entry in the MEASUREMENT EXPLORER window (Figure 102), if active zones are defined on the respective controller.

Status	Tmax / °C	Tavg / °C	Tmin / °C	Index	Fibre	Name	Start	End	Alarm	Pre-alarm	Fault
1	21,60	20,95	20,63	1	Fibre 1	Zone 1	0,00	100,00	-	-	No
2	20,76	20,61	20,41	2	Fibre 1	Zone 2	100,00	200,00	No	-	No
3	20,50	20,34	20,14	3	Fibre 1	Zone 3	200,00	300,00	No	-	No
4	20,36	20,12	19,97	4	Fibre 1	Zone 4	300,00	400,00	No	-	No
5	20,17	20,07	19,95	5	Fibre 1	Zone 5	400,00	500,00	No	No	No
6	20,12	19,88	19,68	6	Fibre 1	Zone 6	500,00	600,00	-	-	No
7	19,86	19,69	19,51	7	Fibre 1	Zone 7	600,00	700,00	-	-	No
8	19,60	19,45	19,29	8	Fibre 1	Zone 8	700,00	800,00	-	-	No
9	19,39	19,19	19,00	9	Fibre 1	Zone 9	800,00	900,00	-	-	No
10	19,29	19,11	18,94	10	Fibre 1	Zone 10	900,00	1000,00	-	-	No
11	18,98	18,78	18,67	11	Fibre 1	Zone 11	1000,00	1100,00	-	-	No
12	18,91	18,65	18,43	12	Fibre 1	Zone 12	1100,00	1200,00	-	-	No
13	18,57	18,42	18,29	13	Fibre 1	Zone 13	1200,00	1300,00	-	-	No
14	18,57	18,18	17,99	14	Fibre 1	Zone 14	1300,00	1400,00	-	-	No
15	18,27	18,11	17,97	15	Fibre 1	Zone 15	1400,00	1500,00	-	-	No
16	18,00	17,87	17,67	16	Fibre 1	Zone 16	1500,00	1600,00	-	-	No

Figure 123: Zone view window

The ZONE DATA window shows the state, index, fibre, name, starting and end position, presence of alarms, pre-alarms and faults as well as maximum, average and minimum temperatures for all zones. Green indicates the normal state, yellow a fibre fault, orange a pre-alarm and red an alarm state of the zone. You can filter the zone entries using the fibre and events filters in the left part of the win-



dow. Zone entries are sorted according to the content of any of the columns in the right part of the window by clicking on the header of the respective column.

The content of the ZONE DATA window corresponds to the state of the controller after the last measurement respectively the measurement activated in the temperature profile window. So far, a reset does not update the zone view.



8 Data Handling

All controller and measuring data is saved in the Charon3 database on your computer. You can import and export data using the entries in the START menu (Figure 125) or in the MEASUREMENT EXPLORER menu (Figure 127).

8.1 Import

Measurement data and controller configurations previously exported from databases of any other Charon3 software installation with the same or lower software version can be imported to the database of the current installation using the IMPORT button of the START menu (Figure 124). Additional controller licenses can also be loaded using the IMPORT button. The OPEN A MEASUREMENT DATA FILE window (Figure 125) opens after pressing the IMPORT button. File names of measurement data files have the extension .mex, controller configuration files have the extension *.cg3 and controller licenses have the extension *.li3. All data formats are intended for sole use by Charon3, and not for use by other software. Select the folder and file name of the data to be imported and press OPEN to start the import. The import of controller licenses is only possible if the controller is connected.

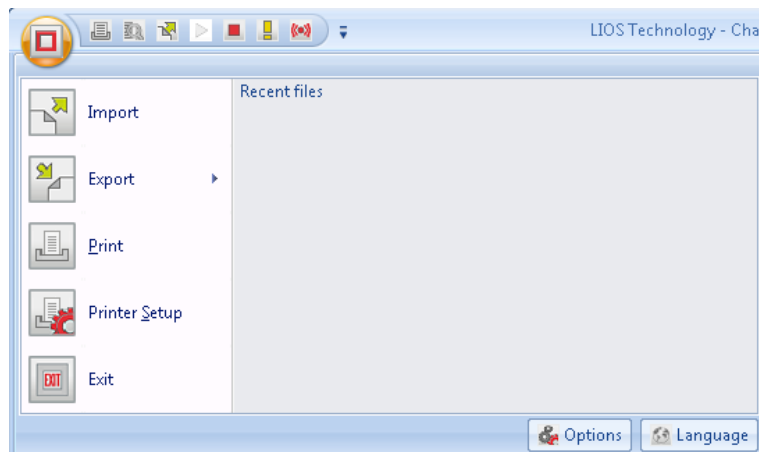


Figure 124: Start menu

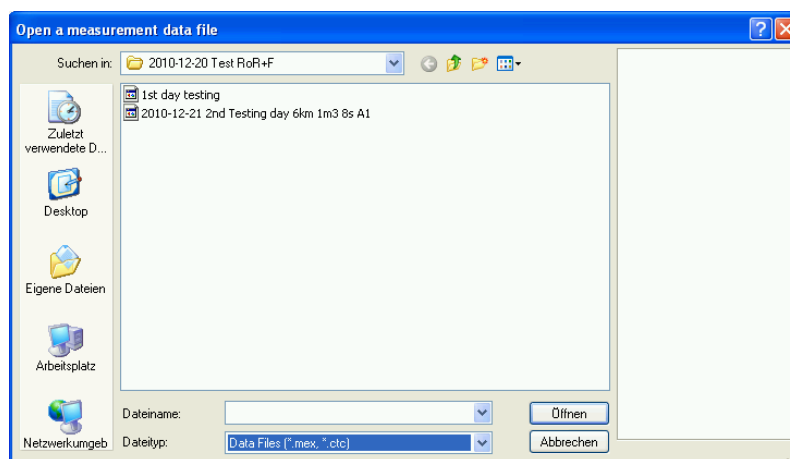


Figure 125: Open a measurement data file window

8.2 Export

Measurement data, zone temperature and controller configurations can be exported from the Charon3 database to separate data files using the EXPORT button of the START menu (Figure 126) or the MEASUREMENT EXPLORER menu (Figure 127). File names of measurement data files for use with



other Charon3 instances have the extension *.mex, whereas controller configuration files have the extension *.cg3.

Measurement data can also be exported to ASCII (*.txt) files in order to be used by other data analysis software.

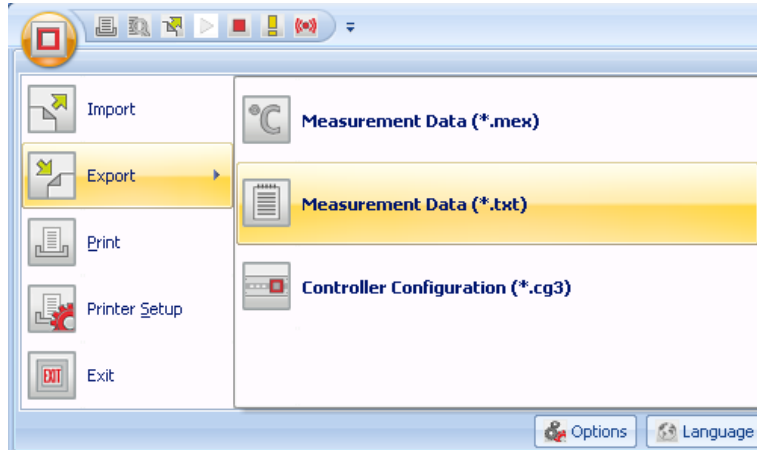


Figure 126: Export menu in the start menu

The export of a measuring session as a *.mex file is also available in the measurement explorer window. A right mouse click on the session opens a pull down menu with the EXPORT entry (Figure 127).

The EXPORT window (Figure 128) opens after selecting measurement data export as a *.mex file. You can choose type and time range of data before starting the export.

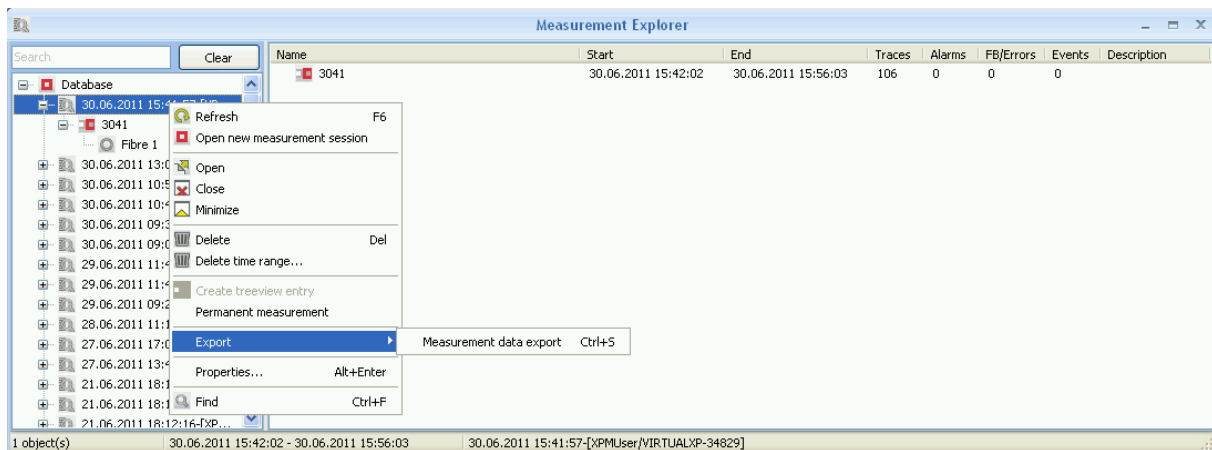


Figure 127: Measurement data export in the measurement explorer

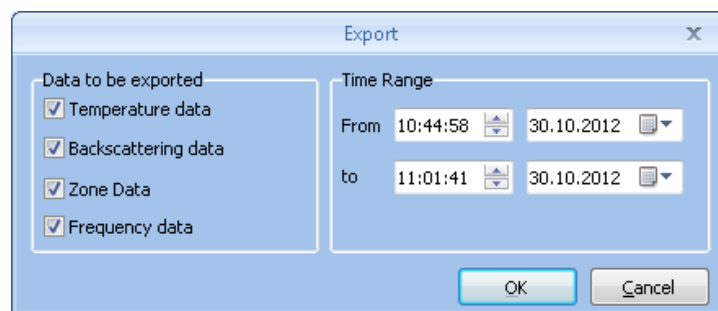


Figure 128: Selection of measurement data to be exported as *.mex file



The ASCII EXPORT window (Figure 129) opens after selecting a temperature window or a zone data window and using the EXPORT button of the START menu. Select measurement data export as a *.txt file (Figure 126).

For exporting temperature data you can choose various options on the data format, type of data and header as shown in Figure 129. You can also select certain ranges of temperature profile (trace) and data point numbers. Furthermore, time and/or distance intervals of the exported data can be increased by selecting the export of each xth trace or data point only. This option is quite useful to considerably reduce the amount of exported data.

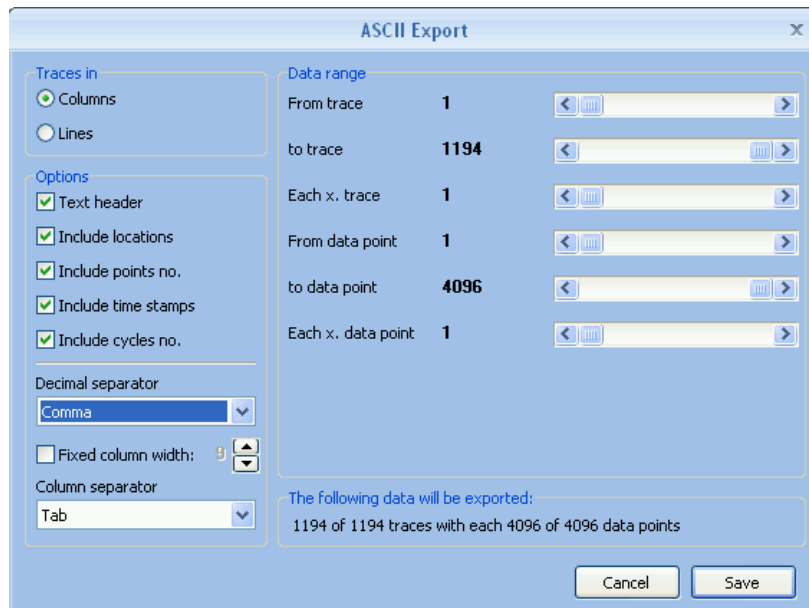


Figure 129: Settings for measured temperature data to be exported in ASCII format

For exporting zone data you can choose various options on the data format, type of data and header as shown in Figure 130.

You can select

- certain ranges of zone temperature data (traces),
- the zone maximum, minimum and average temperature, and
- zones for which the data shall be exported.

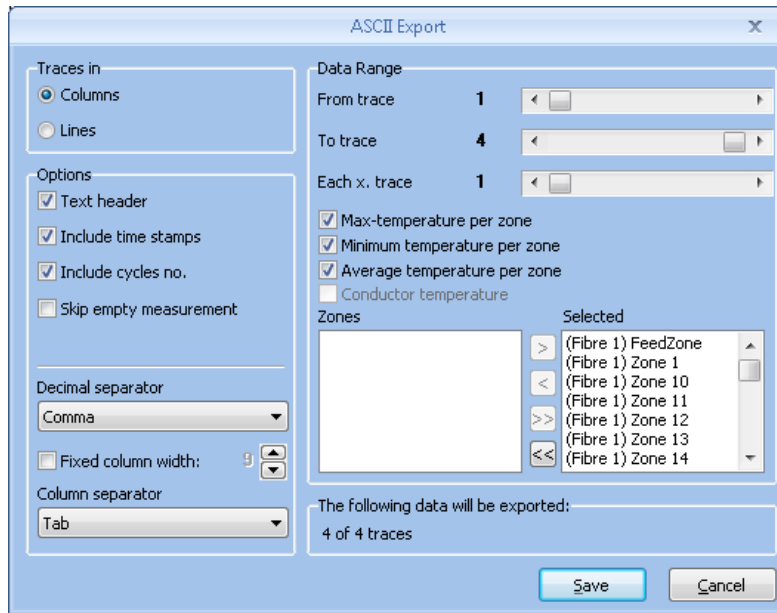


Figure 130: Settings for zone temperature data to be exported in ASCII format



9 Parameter Sets

9.1 EN 54-22 Approved Parameter Sets

The measurement parameter sets shown in Table 10 are approved according to EN 54-22. Parameter sets matching the number of channels of the controller and the length of the connected fibres are displayed in the measurement parameter window (Figure 63) and can be selected. The available distance range is further limited by the distance range of the used controller model (Table 11).

Table 10: EN54-22 measurement parameter sets

Name	# of Channels	Sampling Intervall	Measuring Time per Channel	Measuring Time
		m	s	s
EN 54-22 1CH 3m	1	3	16	16
EN 54-22 1CH 2m	1	2	16	16
EN 54-22 1CH 1m	1	1	16	16
EN 54-22 1CH 0,5m	1	0,5	16	16
EN 54-22 1CH 0,25m	1	0,25	16	16
EN 54-22 2CH 3m	2	3	8	16
EN 54-22 2CH 2m	2	2	8	16
EN 54-22 2CH 1m	2	1	8	16
EN 54-22 2CH 0,5m	2	0,5	8	16
EN 54-22 2CH 0,25m	2	0,25	8	16
EN 54-22 4CH 3m	4	3	4	16
EN 54-22 4CH 2m	2	2	4	16
EN 54-22 4CH 1m	4	1	4	16
EN 54-22 4CH 0,5m	4	0,5	4	16

Table 11: Maximum sensor cable length of the LHD3 and LHD3S controller models

Name	LHD3	LHD3- EXT3	LHD3- EXT4	LHD3S	LHD3S- EXT3	LHD3S- EXT4
	km	km	km	km	km	km
EN 54-22 1CH 3m	14	10	4	5	5	2
EN 54-22 1CH 2m	9	9	2	4	4	2
EN 54-22 1CH 1m	4,5	4	-	2	2	1
EN 54-22 1CH 0,5m	2	2	-	1	1	-
EN 54-22 1CH 0,25m	1	-	-	-	-	-
EN 54-22 2CH 3m	14	10	4	4	4	2
EN 54-22 2CH 2m	9	9	2	4	4	2
EN 54-22 2CH 1m	4,5	4	-	2	2	1
EN 54-22 2CH 0,5m	2	2	-	1	1	-
EN 54-22 2CH 0,25m	1	-	-	-	-	-
EN 54-22 4CH 3m	6	6	2	4	4	2
EN 54-22 4CH 2m	4	4	-	3	3	1
EN 54-22 4CH 1m	2	2	-	2	2	-
EN 54-22 4CH 0,5m	1	1	-	1	1	-



The alarm parameter sets shown in Table 12 are approved according to EN 54-22. In projects which require the compliance with EN54-22 only these approved alarm parameter sets considering the response class have to be used.

Table 12: EN 54-22 alarm parameter sets

Name	Response Class	Tmax	Tmin	ΔT1	Δt1	ΔT2	Δt2	ΔT3	Δt3	Hot Spot ΔT	Delay
		°C	°C	°C	s	°C	s	°C	s	°C	Cycles
EN 54-22 A1N	A1N	60	-	20.8	96	24	160	24	320	-	1
EN 54-22 BN	BN	77	-	-	-	-	-	-	-	-	1
EN 54-22 CN	CN	92	-	-	-	-	-	-	-	-	1

Caution!

Both, EN 54-22 measurement and EN 54-22 alarm parameter sets must be used to operate the controller in compliance with EN 54-22.

9.2 UL521 / ULC S530 Approved Parameter Sets

The measurement parameter sets shown in Table 13 are approved according to UL521 / ULC S530 for the LHD3 controller series, but not for the LHD3S controller series. Parameter sets matching the number of channels of the controller and the length of the connected fibres are displayed in the measurement parameter window (Figure 63) and can be selected. The available distance range is further limited by the distance range of the used controller model.

Table 13: UL521 / ULC S530 measurement parameter sets

Name	# of Channels	Maximum Distance Range	Sampling Interval	Measuring Time	Cycle Time
		km	m	s	s
UL521 1CH 3m	1	10	3	16	16
UL521 1CH 1m	1	4	1	16	16
UL521 1CH 0.5m	1	2	0,5	16	16
UL521 2CH 3m	2	10	3	8	16
UL521 2CH 1m	2	4	1	8	16
UL521 2CH 0.5m	2	2	0.5	8	16
UL521 4CH 3m	4	6	3	4	16
UL521 4CH 1m	4	2	1	4	16

The alarm parameter sets shown in Table 14 are approved according to UL521 / ULC S530.



Table 14: UL521 / ULC S530 alarm parameter sets

Name	Tmax	$\Delta T1$	$\Delta t1$	$\Delta T2$	$\Delta t2$	$\Delta T3$	$\Delta t3$	Hot Spot ΔT	Delay
	°F (°C)	°F (°C)	s	°F (°C)	s	°F (°C)	s	°F (°C)	Cycles
UL521 Fixed 135F	135 (57.2)	-	-	-	-	-	-	-	1
UL521 Fixed 160F	160 (71.1)	-	-	-	-	-	-	-	0
UL521 Fixed 190F	190 (87.8)	-	-	-	-	-	-	-	1
UL521 ROR 15F / 20F	-	16.2 (9)	315	-	-	-	-	-	1

The ratings of the spacing and the maximum installation temperature according to UL / ULC are shown in Table 15.

Table 15: Ratings of the spacing and the maximum installation temperature

Name	Maximum Spacing	Maximum distance from the wall	Maximum installation temperature of the sensor cable
	ft (m)	ft (m)	°F (°C)
UL521 Fixed 135F	50 (15.24)	25 (7.62)	100 (37.8)
UL521 Fixed 160F	30 (9.14)	15 (4.57)	100 (37.8)
UL521 Fixed 190F	50 (15.24)	25 (7.62)	150 (65.6)
UL521 ROR 15F / 20F	50 (15.24)	25 (7.62)	150 (65.6)

Caution!

Compliance to UL / ULC is obtained only if parameter sets shown in Table 13 and Table 14 are used in conjunction with the approved sensor cable types “Fast Response” and “Metal Free”.

9.3 Other Parameter Sets

The measurement parameter sets shown in Table 16 are not approved. They may be used in applications requiring very fast heat detection, but not requiring an approval. You may consider using an extra fast responding sensor cable in applications requiring very fast detection. Please ask the LIOS Sales for appropriate sensor cables for special applications.



Warning!

Since measurement and alarm parameters are crucial for fire detection response of the controller, only the approved parameter sets must be used to ensure operation compliant to European, North American and national standards such as EN 54-22 and UL521 / ULC S530. The use of any other measurement or alarm parameter sets is not compliant to the EN 54-22 and UL521 / ULC S530 approval!

Parameter sets matching the number of channels of the controller and the length of the connected fibres are displayed in the measurement parameter window (Figure 63) and can be selected. The available distance range is further limited by the distance range of the used controller model.



Table 16: Other measurement parameter sets

Name	# of Channels	Maximum Distance Range	Sampling Interval	Measuring Time	Cycle Time
		km	m	s	s
Fast 1CH 3m	1	14	3	8	8
Fast 1CH 1m	1	4.5	1	8	8
Fast 1CH 0.5m	1	2.2	0,5	8	8
Fast 2CH 5m	2	10	5	4	8
Fast 2CH 3m	2	6.5	3	4	8
Fast 2CH 1m	2	2.2	1	4	8
Fast 2CH 0.5m	2	1	0,5	4	8
Ultrafast 1CH 3m	1	6.5	3	4	4
Ultrafast 1CH 1m	1	2.2	1	4	4
Special 1CH 0.5m	1	1	0,5	4	4
Special 1CH 0.25m	1	1	0,25	8	8

The alarm parameter sets shown in Table 17 are intended for applications not requiring an approval according to EN 54-22 or UL521 / ULC S530.

The TUNNEL alarm parameter sets are intended for use in non-EN 54-22 and non-UL521 tunnel applications and in combination with FAST measuring parameter sets. The TUNNEL STANDARD alarm parameter set can be used inside a tunnel, whereas the less sensitive TUNNEL PORTAL parameter set can be used for tunnel portals. Do not use the tunnel alarm parameter sets in combination with other measuring parameter sets.

The PRE-ALARM and TESTMODE alarm parameter sets are intended for triggering pre-alarms and for testing using the FAST measurement parameter sets. Due to their enhanced sensitivity, these parameter sets may not be suited for alarm triggering in permanent heat detection.

The FAST and ULTRAFast alarm parameter sets are intended for use in combination with the FAST and ULTRAFast measuring parameter sets, respectively. Do not use the FAST and ULTRAFast alarm parameter sets in combination with other measuring parameter sets.

Table 17: Other alarm parameter sets

Name	Tmax	Tmin	$\Delta T1$	$\Delta t1$	$\Delta T2$	$\Delta t2$	$\Delta T3$	$\Delta t3$	Hot Spot ΔT	Delay
	°C	°C	°C	s	°C	s	°C	s	°C	Cycles
Tunnel Standard	55	-	7	57	9	105	12	153	15	1
Tunnel Portal	60	-	11	57	12	105	14	153	20	1
Pre-alarm	45	-	6	57	8	105	10	153	-	-
Testmode	40	-	5	57	8	105	10	153	8	-
Fast	50	-	6	41	9	105	12	153	10	1
Ultrafast	50	-	5	29	9	53	12	81	10	1

9.4 Parameter Sets for EX Controllers

Basically, the measurement and alarm parameter sets shown in



Name	# of Channels	Sampling Interval	Measuring Time per Channel	Measuring Time
		m	s	s
EN 54-22 1CH 3m	1	3	16	16
EN 54-22 1CH 2m	1	2	16	16
EN 54-22 1CH 1m	1	1	16	16
EN 54-22 1CH 0,5m	1	0,5	16	16
EN 54-22 1CH 0,25m	1	0,25	16	16
EN 54-22 2CH 3m	2	3	8	16
EN 54-22 2CH 2m	2	2	8	16
EN 54-22 2CH 1m	2	1	8	16
EN 54-22 2CH 0,5m	2	0,5	8	16
EN 54-22 2CH 0,25m	2	0,25	8	16
EN 54-22 4CH 3m	4	3	4	16
EN 54-22 4CH 2m	2	2	4	16
EN 54-22 4CH 1m	4	1	4	16
EN 54-22 4CH 0,5m	4	0,5	4	16

Table 11: Maximum sensor cable length of the LHD3 and LHD3S controller models

to Table 14, and Table 16 and Table 17 are also available for EXT3 and EXT4 controller models. However, these controller models are not approved according to EN 54-22 or UL521 / ULC S530. Furthermore, the measurement performance of these controller models is restricted by the lower laser power permitted in the explosive environments. Therefore, the useful distance range of the measurement parameter sets may be lower than for the non-EX controller models (Table 18).

Table 18: Useful distance ranges for EX controller models

Measurement Parameter Set	Pa- Channels	# of Channels	Sampling Interval	Measuring Time	Cycle Time	Useful Distance Range EXT3	Useful Distance Range EXT4
			m	s	s	km	km
EN 54-22 1CH 3m		1	3	16	16	10	10
EN 54-22 1CH 1m		1	1	16	16	4.5	4.5
EN 54-22 1CH 0.5m		1	0,5	16	16	2	2
EN 54-22 1CH 0.25m		1	0,25	16	16	1	-
EN 54-22 2CH 3m		2	3	8	16	10	6
EN 54-22 2CH 1m		2	1	8	16	4.5	2
EN 54-22 2CH 0.5m		2	0,5	8	16	2	1
EN 54-22 2CH 0.25m		2	0,25	8	16	1	-
EN 54-22 4CH 3m		4	3	4	16	6	4
EN 54-22 4CH 1m		4	1	4	16	2	1
EN 54-22 4CH 0.5m		4	0,5	4	16	1	-
Fast 1CH 3m		1	3	8	8	10	6
Fast 1CH 1m		1	1	8	8	4.5	2
Fast 1CH 0.5m		1	0,5	8	8	2.2	1
Fast 2CH 5m		2	5	4	8	10	6
Fast 2CH 3m		2	3	4	8	6.5	4



Fast 2CH 1m	2	1	4	8	2.2	1
Fast 2CH 0.5m	2	0,5	4	8	1	-
Ultrafast 1CH 3m	1	3	4	4	6.5	4
Ultrafast 1CH 1m	1	1	4	4	2.2	1
Special 1CH 0.5m	1	0,5	4	4	1	-
Special 1CH 0.25m	1	0,25	8	8	1	-

10 Computer Requirements and Software License

10.1 Computer Requirements

10.1.1 Hardware Requirements

The computer software for installation and operation of the controller runs reliably on computers with the minimum requirements listed in Table 19. Different hardware with comparable performance can also be used. More powerful computers can be used to accelerate processing and for saving larger amounts of measuring data.

About 250 million data points can be stored on 1 GB disk space. For example, saving temperature profiles of a continuous measurement with 10 km distance range, 1 m sampling interval and 5 minutes measuring time requires about 20 MB of disk space per day. Saving of backscattering curves requires even more disk space.

Table 19: Hardware requirements

<i>Program</i>	<i>Application</i>	<i>Hardware Requirement</i>
Charon3	Commissioning	1.3 GHz processor, e.g. Pentium 4 512 MB RAM memory 250 MB free hard disk space 1024x600 display resolution
Charon3	Data storage, visualization	2 GHz dual core processor 1024 MB RAM memory 1 GB free hard disk space 1024x768 (VGA) display resolution 19" screen Uninterruptible power supply (UPS)
Real-Time Thermal Rating (RTTR) plug-in (option)	Thermal modelling of power cables	2 GHz quad core processor 2048 MB RAM memory 1 GB free hard disk space 1024x768 (VGA) display resolution 19" screen Uninterruptible power supply (UPS)

10.1.2 Operating System

Charon3 software can only be installed on computers using one of the following 32-bit operating systems:

Table 20: Windows operating system requirements



<i>Operating System</i>	<i>Minimum Service Pack</i>
Windows 7	SP 1
Windows Vista	SP 2
Windows XP	SP 3

10.2 Software License

10.2.1 Subject of the Contract

LIOS Technology GmbH (hereinafter referred to as "LIOS"), grants its customer (hereinafter referred to as the "User") a non-exclusive and non-transferable right to use any software delivered to the User by LIOS (hereinafter referred to as the "Software") under the following contractual terms and conditions (hereinafter referred to as the "License"). Unless otherwise agreed upon in writing these terms shall apply to software delivered by LIOS to the User, notwithstanding if LIOS or a third party has developed/produced the software.

When accepting these terms and conditions the User accepts and acknowledges that in case of delivery of software from a third party developer/producer the rights and obligations of these terms and conditions apply mutatis mutandis to such third party.

10.2.2 Limited Warranty

For a period of 6 months following delivery of the Software to the User (the "Warranty Period"), LIOS warrants that the Software shall perform in all material respects according to LIOS' specifications concerning the Software when used with the appropriate equipment. In the event of any breach or alleged breach of this warranty, the User shall promptly notify LIOS and return the Software to LIOS at User's expense. The User's sole remedy shall be that the User shall correct the Software so that it operates according to the warranty. This warranty shall not apply to the Software if modified by the User or if used improperly or on an operating environment not approved by LIOS.

LIOS hereby represents and warrants to User that LIOS is the owner of the Software or otherwise has the right to grant to User the rights set forth in this License. In the event of any breach or threatened breach of the foregoing representation and warranty, User's sole remedy shall be to require LIOS or to either:

- i) procure, at LIOS' expense, the right to use the Software,
- ii) replace the Software or any part thereof that is in breach and replace it with Software of comparable functionality that does not cause any breach, or
- iii) refund to User the full amount of the fee payable under the License upon the return of the Software and all copies thereof to LIOS.

Such remedies apply only after User in writing notifying LIOS of the breach and allowing LIOS a period of 30 days to rectify the breach in matter.

LIOS' warranties set forth in this License are exclusive and are in lieu of all other warranties, express or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.



10.2.3 Liability

In no event shall LIOS' liability hereunder exceed the amount of fees paid by the User to LIOS under the current project or agreement etc., regardless of whether User's claim is based on contract, tort, strict liability, product liability or otherwise.

LIOS shall not be liable for any indirect or consequential damages, such as lost profit, cost-savings which fail to materialize, damages arising from the claims of third parties or loss of data etc.

LIOS shall not be liable for the accuracy of information or particulars passed to third parties by telephone.

10.2.4 Safeguard against Misuse

Title to all present and future intellectual property rights such as copyright and/or commercial patents to the Software licensed and to all programs derived there from, sections of programs or supporting documentation prepared in this context remains with LIOS.

10.2.5 Open Source Licenses

The controller uses software code developed by third parties and provided under the terms of the GNU General Public License (GPL)⁷, the GNU Lesser General Public License (LGPL)⁸ or other Open Source licenses. You can obtain the source code of the third party software including changes made by LIOS Technology on a CD for a fee covering the medium, handling and shipment. Contact Technical Support (see section 12.2.4) to obtain the CD.

The CD is offered for legal reasons. You do not need the CD to operate the controller.

10.2.6 Other Restrictions

Except as expressly authorized above, User shall not: copy, in whole or in part, Software or documentation; modify the Software; reverse compile or reverse assemble all or any portion of the Software; or rent, lease, distribute, sell, or create derivative works of the Software.

This License will terminate immediately without notice from LIOS if the User fails to comply with any provision of this License.

10.2.7 Governing Law

This Agreement shall be construed and enforced in accordance with the laws of the Federal Republic of Germany, however not applying its rules regarding choice of law.

10.2.8 No Assignment

Neither this License nor any interest in this License can be assigned by User without the prior express written approval of LIOS.

⁷ See <http://www.gnu.org/licenses/gpl.html>

⁸ See <http://www.gnu.org/licenses/lgpl.html>



11 Troubleshooting

11.1 General

The controller is basically maintenance-free. The following section gives explanations of alarm, fault, warning and information messages as well as the default assignment of address points sent by the controller or saved in the event memory. It also gives recommendations for solving possible issues and for contacting the technical support.

11.2 Fibre Address Points

Table 21: Fibre address point assignments of the LON protocol in the default (OTS3) mode

Event	Fibre Address Point Assignment	Fibre Address Point Range
Alarm in zone i (i = 1 ... 1000)	i	1 to 1000
Pre-alarm or control in zone i	i + 1000	1001 to 2000
Fibre-break in zone i	i + 2000	2001 to 3000
Fire magnitude j (j = 1 ... 5)	j + 3000	3001 to 3005
Fire static	3010	3010
Fire moving towards fibre end	3011	3011
Fire moving towards controller	3012	3012

Caution!

The assignments of fibre and system address points shown in Table 21 to Table 24 are not used in the non-default compatibility modes of the LON protocol. Fibre address points need to be assigned individually if 5505, 2005 or 1056 LON protocol versions are used (Table 8).

11.3 Errors

Table 22: Error messages and recommendations for troubleshooting

Name	System Address Point	Description	Troubleshooting
System fault	101	Triggered by any controller fault, fibre break, measurement stopped or power off.	Check further error messages to analyse the issue.
Configuration Error	102	Configuration is not loaded by the controller. Maybe out of memory.	Repeat commissioning with reduced number of zones and/or use alarm parameter sets with shorter time intervals. If necessary, send diagnosis data to technical support.
Fibre break	103	Unexpected change of the sensor cable length.	Check the sensor cable for fibre breaks and repair it (see Installation Manual).
Measurement error	104	Invalid measurements. Maybe the fibre was broken or disconnected during measurement.	Check fibre. Repeat measurement and/or commissioning. If necessary, send diagnosis data to technical support.
Device error, call support	105	Controller hardware error requiring support.	Send diagnosis data to the technical support (see section 6.3.5)
Device error	106	Controller error, e.g. low	Switch the controller off. Change



		buffer battery.	buffer battery and repeat commissioning if the error description indicates low battery. Switch the controller on. If necessary, send diagnosis data to technical support.
Environmental conditions error	107	Supply voltage, temperature and/or humidity are out of range.	Provide proper environmental conditions. Improve air flow around controller.
Handling error	108	No fibre, ...	Check controller configuration.
Measurement stopped	109	Message sent once when the measurement is stopped.	Start measurement if required.
Measurement inactive	110	Periodic message when the measurement is inactive.	Start measurement if required.

11.4 Warnings

Table 23: Warnings and recommendations for troubleshooting

Name	System Address Point	Description	Troubleshooting
Configuration warning	201	Configuration data is not valid.	Repeat commissioning. If necessary, send diagnosis data to technical support.
Measurement warning	202	Single invalid measurement. This measurement is ignored, and measurements are continued.	Check environmental conditions, sensor cable and/or repeat commissioning if the warning appears frequently. If necessary, send diagnosis data to technical support.
Environmental conditions warning	203	Supply voltage, temperature and/or humidity are close to the limits.	Provide proper environmental conditions. Improve air flow around controller.
Communication warning	204	Low quality of data transmission.	Check type and lengths of communication cables as well as network settings.

11.5 Messages

Table 24: Messages for information

Name	System Address Point	Description
Acknowledgement	51	Acknowledgement has been triggered by key switch, Charon3, communication protocol or input.
Reset	52	Reset has been triggered by key switch, Charon3, communication protocol or input.
Test mode activated	53	Test mode has been activated or deactivated. Simulations of inputs, outputs, fibre breaks and/or alarms.
Test mode deactivated	54	Test mode has been deactivated.
Firmware upload	-	A new firmware version has been uploaded to the controller.
System start	-	System has been started.



Output	-	Output state has been changed.
Inputs 1 to 40	1 to 40	Alarm, reset or acknowledgement input has been activated. Control input has been activated or deactivated.

11.6 Network Ports

The controller and Charon3 use UDP broadcast for the discovery of controllers and for configuring network interfaces as well as TCP-based protocols for data transmission. Open the required network ports (Table 25) on your computer and the network server or deactivate the firewalls to enable communication.

Table 25: Network ports

Port	Protocol	Usage
9973/udp	LIOS discover	Discovery of controllers and network configuration
9974/udp	LIOS discover	Discovery of controllers and network configuration
502/tcp	MODBUS/TCP	Optional data transmission to third party systems
2404/tcp	IEC 60870-5-104	Optional data transmission to third party systems
5000/tcp	LON	Data transmission to third party systems
9973/tcp	LIOS communication	Data transmission Charon3 / controller
20000/tcp	DNP3 over LAN/WAN	Optional data transmission to third party systems



12 Information Scheme and Contact Information

12.1 Information Scheme

This user manual forms part of the LHD3 documentation. The following provides an overview of the information scheme for LHD3:

- Datasheets on LHD3 controller models: List the technical data.
- Installation Manual OTS3 / LHD3: Describes the basics of installation of the controller.
- Operation Manual LHD3: Describes basic principles of application of LHD3 controllers in linear heat detection.
- Communication Protocols Manuals: Describes the communication protocols.
 - Part I: LON
 - Part II: DNP3
 - Part III: IEC 60870-5-104
 - Part IV: MODBUS
 - Part V: XML
- Declaration of Conformity: Officially declares compliance with the applicable Directives of the European Commission and the respective European safety standards.

12.2 Contact Information

12.2.1 Online Availability

Updated technical documentation on LHD3 controllers, certificates and software are available for registered customers at the following Internet address:

<https://www.lios-support.com>

Please ask the Technical Support (see section 12.2.4) for registration details.

12.2.2 Representatives and Offices

If you have any further questions relating to products described in this manual, please contact your local representative at the office nearest you.

You can locate your contact partner at this Internet address: <http://www.lios-tech.com>

12.2.3 Training

LIOS Technology offers a variety of training courses to familiarize you with LHD3 controller systems. Please contact your local representative or the Technical Support (see section 12.2.4).

12.2.4 Technical Support

You can reach the Technical Support for all LIOS Technology products using

e mail support@lios-tech.com

Phone +49-221-99887-555 or

FAX +49-221-99887-150.



12.2.5 Contact Information

Table 26: Corporate Head Office Germany

Company	<i>LIOS Technology GmbH</i>
Street	Schanzenstrasse 39 Building D9-D13
City	51063 Cologne
Country	Germany
Phone	+49 221 99887 0
FAX	+49 221 99887 150
e-mail	info@lios-tech.com
Internet URL	http://www.lios-tech.com

Table 27: Sales and Support America

Company	<i>LIOS Technology Inc.</i>
Street	1400 Campus Drive West
City	Morganville, New Jersey 07751
Country	United States of America
Phone	+1 (732) 970 8062
FAX	+1 (732) 972 4410
e-mail	info@lios-tech.com
Internet URL	http://www.lios-tech.com

Table 28: Sales and Support Asia / Pacific

Company	<i>LIOS Technology GmbH</i>
Street	25/F China Fortune Tower 1568 Century Avenue
City	Shanghai 200122
Country	People's Republic of China
Phone	+86 21 6875 6002
FAX	+86 21 6875 8573
e-mail	info@lios-tech.com
Internet URL	http://www.lios-tech.com



Figures

Figure 1: Fibre length of 500 m divided into 5 zones. An alarm is triggered in zone 3 by elevated temperatures around 250_m.....	8
Figure 2: Pre-alarm and alarm triggering at maximum temperatures.....	10
Figure 3: Alarm triggering at a hot spot temperature difference	10
Figure 4: Alarm triggering at a rate-of-rise.....	11
Figure 5: Two-step control of an output using an activation temperature of 50°C and a deactivation temperature of 40°C.....	11
Figure 6: Preparing to install message	16
Figure 7: Welcome window of the InstallShield Wizard	17
Figure 8: License agreement window.....	17
Figure 9: Selection of destination folder	17
Figure 10: Selection of the database folder	18
Figure 11: Ready-to-Install window	18
Figure 12: Error - another installation is in progress.....	19
Figure 13: Confirmation of cancelling the installation	19
Figure 14: Prepare to install	19
Figure 15: User Account Control (UAC) on Windows operating systems with UAC	20
Figure 16: Windows 7 configuration of Charon03	20
Figure 17: UAC request on Windows operating systems with UAC.....	20
Figure 18: UAC Message on Windows operating system with UAC.....	20
Figure 19: Progress bar during Charon3 installation.....	21
Figure 20: Attach device hint	21
Figure 21: Finish installation window.....	21
Figure 22: Charon3 entry in the START menu of Windows.....	21
Figure 24: Charon3 window in the user level 1.....	22
Figure 25: Charon3 window in user level 3	23
Figure 26: Tree view with pull down menu of a controller.....	24
Figure 27: Confirmation of update.....	25
Figure 28: Information that a downgrade is not possible.....	25
Figure 29: Window messages after connection to a controller using USB	25
Figure 30: Welcome message of the Found New Hardware Wizard	26
Figure 31: Selection of installation mode.....	26
Figure 32: Selection of driver location	26
Figure 33: Hardware installation wizard.	27
Figure 34: Finish hardware installation wizard.	27
Figure 35: Hardware installed	27
Figure 36: DEVICE tab with QUICK START button on the left.....	28
Figure 37: START window with message SEARCHING FOR DEVICES (being displayed if no controller is connected).....	29
Figure 38: START window with CONTROLLER TABLE being displayed if controller(s) are connected ..	29
Figure 39: TCP/IP properties windows with (left) and without (right) warning sign	30
Figure 40: Network menu - DNS/NTP window.....	32
Figure 42: Controller window.....	33
Figure 43: SCAN ALL and STOP SCANNING.....	34
Figure 44: Scanning a single channel.....	34



Figure 45: Applications window	35
Figure 46: Tree View after finishing the QUICKSTART.....	35
Figure 47: Alarm parameter set tab of the alarm / control parameter sets window	38
Figure 48: Control parameter set tab.....	38
Figure 50: Event configuration, general tab.....	40
Figure 51: External Actions, Faults & Warnings tab.....	41
Figure 52: Message: System address points are set to default values.....	42
Figure 53: Inputs tab	42
Figure 54: Outputs tab	43
Figure 55: Communication settings: LON protocol	45
Figure 56: LON protocol change warning.....	46
Figure 58: Configuration of the data transmission over the XML protocol	48
Figure 59: Locking the controller.....	49
Figure 60: Warning before locking the controller	49
Figure 61: Display of a locked controller in the TREE VIEW (locker icon)	49
Figure 62: Measurement sequence window.....	50
Figure 63: Measurement parameter window with pre-defined measurement parameter sets.....	51
Figure 64: Temperature Calibration window	52
Figure 65: Zone configuration window	53
Figure 66: Shift selected zones window	54
Figure 67: Generate zones window.....	54
Figure 68: Fibre events window	56
Figure 69: Length scaling window	57
Figure 70: Temperature profile with the scaled section 2	57
Figure 71: Different configurations on controller and PC window	58
Figure 72: Task bar of the DEVICE tab.....	59
Figure 73: Tree view during warm-up (left) and running measurement (right)	59
Figure 74: BACKSCATTERING window	60
Figure 75: BACKSCATTERING window: Determination of the attenuation coefficient.....	61
Figure 76: BACKSCATTERING windows showing curves indicating proper (left) and improper (right) optical termination of the fibre.....	62
Figure 77: Service tab of Charon3	63
Figure 79: Data download window	64
Figure 82: Tree view of a controller in the test mode.....	66
Figure 83: Test mode section of the service tab after activation of the test mode.....	67
Figure 84: Perform test window.....	67
Figure 86: Configuration report.....	69
Figure 88: Project information window	71
Figure 89: Support information window	72
Figure 90: Diagnosis Data - General tab after entering the information	72
Figure 91: Diagnosis Data - Status tab.....	73
Figure 92: Diagnosis Data - Description tab	74
Figure 93: Select firmware window.....	74
Figure 94: Firmware license agreement.....	75
Figure 95: Firmware upload confirmation window.....	75
Figure 97: Quick access toolbar.....	76



Figure 98: Toolbar tab of the customize window 77

Figure 99: Options tab of the customize window 77

Figure 100: Edit tab 78

Figure 101: View tab..... 78

Figure 102: Measurement explorer window..... 78

Figure 103: Message window..... 79

Figure 104: Temperature profile and temperature history windows 80

Figure 105: General tab of the options window 81

Figure 106: Default configuration tab of the options window..... 82

Figure 107: Application settings window 83

Figure 109: Application settings window - general tab..... 84

Figure 110: Application settings window - default zones tab 84

Figure 111: Application settings window - measuring parameter sets tab 85

Figure 112: Application settings window - alarm parameter sets tab 86

Figure 114: Measurement sessions tab of the options window 87

Figure 115: Export tab of the options window..... 88

Figure 116: Language window..... 89

Figure 117: Plug-In Window 89

Figure 118: Windows tab 89

Figure 120: Temperature profile window 90

Figure 121: Curve repository in the temperature profile window..... 91

Figure 122: Backscattering profile window..... 92

Figure 123: Zone view window..... 92

Figure 124: Start menu..... 94

Figure 125: Open a measurement data file window..... 94

Figure 126: Export menu in the start menu 95

Figure 127: Measurement data export in the measurement explorer 95

Figure 129: Settings for measured temperature data to be exported in ASCII format 96

Figure 130: Settings for zone temperature data to be exported in ASCII format..... 97



Tables

Table 1: Fire magnitude classes.....	12
Table 2: Fire propagation directions	12
Table 3: User levels.....	23
Table 4: Columns of the CONTROLLER TABLE and their content	29
Table 5: TCP/IP properties settings	31
Table 6: Fibre status values and their interpretation.....	33
Table 7: Alarm parameter sets	36
Table 8: LON protocol versions	44
Table 9: Optional licenses.....	76
Table 10: EN54-22 measurement parameter sets	98
Table 11: Maximum sensor cable length of the LHD3 and LHD3S controller models	98
Table 12: EN 54-22 alarm parameter sets.....	99
Table 13: UL521 / ULC S530 measurement parameter sets	99
Table 14: UL521 / ULC S530 alarm parameter sets.....	100
Table 15: Ratings of the spacing and the maximum installation temperature.....	100
Table 16: Other measurement parameter sets.....	101
Table 17: Other alarm parameter sets	101
Table 18: Useful distance ranges for EX controller models.....	102
Table 19: Hardware requirements	103
Table 20: Windows operating system requirements	103
Table 21: Fibre address point assignments of the LON protocol in the default (OTS3) mode	106
Table 22: Error messages and recommendations for troubleshooting	106
Table 23: Warnings and recommendations for troubleshooting.....	107
Table 24: Messages for information.....	107
Table 25: Network ports.....	108
Table 26: Corporate Head Office Germany	110
Table 27: Sales and Support America.....	110
Table 28: Sales and Support Asia / Pacific.....	110



Abbreviations

CD	Compact disk
CF	Compact Flash
CFR	Code of Federal Regulations
DHCP	Dynamic host configuration protocol
DNS	Domain name system
DTS	Distributed temperature sensing
EC	European Commission
EN	European standard
EX	Explosion (protection)
IEC	International Electronic Commission
IP	Internet protocol
LAN	Local area network
Laser	Light amplification by stimulated emission of radiation
LED	Light emitting diode
LHD	Linear heat detection
LON	Local operating network
MM	Multimode
N.A.	Numerical aperture
NTP	Network time protocol
OTS	Optical temperature sensing
PA	Polyamide
PC	Personal computer
PDF	Portable data format
RAM	Random access memory
rtf	Rich text format
SCADA	Supervisory control and data acquisition
SM	Single-mode



UL	Underwriters Laboratory
UPS	Uninterruptible power supply
USB	Universal serial bus
UT	Universal Time
VGA	Video graphics array

Glossary

<i>Acknowledge</i>	Acknowledging an alarm confirms that you have noted it. The buzzer is deactivated by the acknowledgment.
<i>Alarm delay</i>	Number of subsequent alarming conditions in a zone required for triggering an alarm or pre-alarm.
<i>Alarm parameter set</i>	Set of parameters used to trigger alarms or pre-alarms.
<i>Average temperature</i>	Arithmetic mean of all temperature readings within a zone.
<i>Averaging</i>	Number of subsequently measured temperature profiles used for calculating a time-averaged temperature profile.
<i>Backscattering profile</i>	Set of anti-Stokes or Stokes signal power data in dependence on distance.
<i>Connector position</i>	Distance between the optical port of the controller and an optical connector or splice.
<i>Connector shift</i>	Apparent temperature shift at an optical connector.
<i>Control parameter set</i>	Set of parameters used to control external devices.
<i>Controller</i>	General term for LIOS distributed sensing devices.
<i>Cycle time</i>	Time for measuring a complete sequence of channels.
<i>Distance range</i>	Maximum length of optical fibre to be measured.
<i>EMC</i>	Electromagnetic compatibility is the ability of electrical equipment to function properly in its electromagnetic environment without influencing this environment.
<i>End of T profile</i>	End point of a transmitted temperature profile.
<i>Fibre type</i>	Basic design of an optical fibre.
<i>Maximum fibre length</i>	Maximum fibre length to be measured with the current settings.
<i>Maximum temperature</i>	Maximum temperature in a zone.



<i>Measuring cycle</i>	Procedure of measuring one temperature profile for each fibre in a measurement sequence.
<i>Measuring time</i>	Time required to measure temperatures along a fibre.
<i>Minimum temperature</i>	Minimum temperature in a zone.
<i>Min-Max cycles</i>	Number of measuring cycles used for evaluating the maximum and minimum temperatures. The maximum or minimum temperature at a location is determined by a linear fit to the history of temperature readings.
<i>Output, manual reset</i>	A manual reset output does not automatically deactivate when the activating condition is no longer present. Deactivation requires a reset of the controller. Manual reset outputs can be used for alarm and fault indication.
<i>Output, self-reset</i>	A self-reset output is automatically deactivated when the activating condition is no longer present. Self-reset outputs can be used to control other devices.
<i>Pigtail</i>	A pigtail is a short optical fibre cable with an optical connector on one end and a length of exposed optical fibre at the other end.
<i>Ping</i>	Ping is a computer network utility for testing a connection between computers on an Internet protocol (IP) network.
<i>Reference point</i>	Reference (zero) position for calculating the distance of points in temperature profiles.
<i>Reference temperature</i>	Temperature of a point sensor used as reference for the controller measurement.
<i>Reset</i>	Resetting the controller confirms that events triggering alarms and failures have been properly handled. All alarms, failure states and manual reset outputs are reset.
<i>Return loss</i>	In fibre optics, return loss is the loss of signal power due to a reflection, e.g. caused by a discontinuity of refractive index at a connector, another optical component or the fibre endface.
<i>Sampling interval</i>	Distance between measuring data locations along a fibre.
<i>Send each Xth curve</i>	Number of temperature profiles to be measured between each transmission of one temperature profile.
<i>Spatial resolution</i>	Length of a hot spot required for 90% temperature reading response.
<i>Start of T profile</i>	Starting point of a transmitted temperature profile.
<i>Temperature accuracy</i>	Deviation between average measured temperature and true temperature.



<i>Temperature profile</i>	Set of temperature data in dependence on distance.
<i>Temperature resolution</i>	Standard deviation of local temperature data taken in a measuring sequence.
<i>Warm-up time</i>	Time between power-on and readiness to take measurements according to the specification.
<i>Zone</i>	Range of adjacent locations used for triggering events according to a common alarm or control parameter set.
<i>ΔT (Cold spot)</i>	Temperature difference of a cold spot.
<i>ΔT (Hot spot)</i>	Temperature difference of a hot spot.





This Page Left Intentionally Blank

SAFETY INSTRUCTIONS

OTS3- / LHD3-Series Controllers



Table of Contents

Table of Contents	2
1 Safety Instructions.....	3
1.1 Safety Guidelines.....	3
1.2 General Safety	3
1.3 Electrical Safety	3
1.4 Laser Safety	4
1.5 Application-Specific Safety	6
1.5.1 Fire Detection	6
1.5.2 Railway Tunnels.....	6
1.5.3 Power Cable Monitoring.....	6
1.5.4 Explosive Areas.....	7







1 Safety Instructions

1.1 Safety Guidelines

This manual contains notices you have to consider in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol. Notices referring only to property damage have no safety alert symbol. The notices shown in Table 1 are graded according to the degree and cause of danger.

Table 1: Safety symbols and notices

	Warning!	This symbol and label indicate that death or severe personal injury may result if proper precautions are not taken.
	Warning!	This symbol and label indicate that death or severe personal injury may result from electricity if proper precautions are not taken.
	Warning!	This symbol and label indicate that severe personal injury may result from laser radiation if proper precautions are not taken.
	Warning!	This symbol and label indicate that death or severe personal injury may result from explosion if proper precautions are not taken.
	Caution!	This label without a safety alert symbol indicates that property damage can result if proper precautions are not taken.

1.2 General Safety

Caution!

The OTS3/LHD3-series controller, the sensor cable, splice boxes and accessories are intended to be installed in accordance with the Installation Manual OTS3 / LHD3 that is supplied with the controller. The system must also be installed in a manner that is acceptable to the local inspection authority having jurisdiction.

Switch the controller off before connecting or disconnecting any optical fibre sensor cable. Never pull-off an optical fibre connector from an E2000 optical port during an active measurement. The emitted laser radiation may damage the fibre endfaces of pigtail and controller.

Never connect pigtails with contaminated ferrule endfaces to the controller. Contaminated endfaces may easily be damaged by laser radiation (see Installation Manual OTS3 / LHD3).

1.3 Electrical Safety

Warning!

Electricity can be a danger to health and life. See a doctor immediately after contact with dangerous voltages.

The controller contains components carrying dangerous voltages. Do not open or repair the controller. Do not operate the controller with damaged housing.



Verify that the supply voltage is within the range given at the DC or AC supply socket of the controller before connecting the power supply cable.

The AC power supply cable has to be disconnected before replacing the fuse at the power supply of the controller.

Accumulator terminals may never be shorted or come into contact with the controller housing.

National safety regulations for using electricity must be considered. Electricity supply and earthing must be carried out in accordance with national regulations.

1.4 Laser Safety



Warning!

The controller contains a laser source. Caution, the laser radiation is invisible! The controller is classified as laser class 1M product. The following label is attached close to the fibre connector(s) on the controller:



The laser emits continuously modulated laser radiation at a wavelength of 1550 nm (OTS3/LHD3 models) or 1064 nm (OTS3S/LHD3S models). The modulation frequency is between 500 Hz and 500 MHz. The maximum average power at 1550 nm is 110 mW, and the maximum peak power at this wavelength is 200 mW. The maximum average power at 1064 nm is 37 mW, and the maximum peak power at this wavelength is 70 mW.

The accessible laser radiation represents no danger for the human eye or skin assuming that:

- The radiation is not focussed by optical instruments like magnifying glasses, lenses or microscopes and
- The end of an optical fibre or cable is not approached closer than 10 cm to the eye or skin.

With optical instruments or during direct approach of the optical fibre to the human body, laser radiation can generate heat that damages the skin or the cornea of the eye. Such damages are wet bulges, cataract and burnings of cornea and skin. See a doctor immediately in case of eye damages.

National regulations on laser safety have to be considered during installation and operation of the controller.

The controller must be switched off before directly viewing on fibre ends, splices or optical fibre connectors with or without optical instruments. In addition, the warning label shown in Figure 1 has to be placed close to the mains switch of the controller. Use only CE marked fibre scopes equipped with a 1064 nm / 1550 nm DIR LB3 laser safety filter according to EN 207.



Figure 1: Warning label

Optical fibre ends must not be directly approached to the eye or the skin. Laser radiation emitted from fibre ends must not be focussed with optical instruments onto the eye or the skin.

The controller must be operated with attached fibre(s). Unused E2000 optical fibre ports of the controller must be terminated using dummy connectors.

Emission of laser radiation at the end of a sensor fibre or cable must be prevented by using an optical termination box or a termination kit (see Installation Manual OTS3 / LHD3).

The laser is automatically switched on by starting a measurement. The LEDs designated with "LASER ACTIVE / LASER ON" at the front and rear panels of the controller indicate laser operation.

If an optical fibre sensor cable, pigtail or optical fibre is broken, it must be repaired, terminated or replaced immediately (follow the instructions given in the Installation Manual OTS3 / LHD3).



1.5 Application-Specific Safety

1.5.1 Fire Detection

Caution!

During the commissioning and testing of fire detection controllers (LHD3-series) it must be ensured that:

- false triggering of transmission devices is securely prevented,
- the fire brigade is informed before test triggering of transmission devices,
- triggering of fire extinguishing controls is blocked,
- the responsible person is present when third party devices are triggered,
- the collective fault output of the controller is connected to the alarm control system,
- the collective alarm output of the controller is connected to the alarm control system, and
- a programmable output is assigned to the test mode and connected to the alarm control system if the test mode shall be used.

1.5.2 Railway Tunnels



Warning!

A safe distance from high voltage lines must be maintained in railway tunnels with electrical traction lines. The safety distance depends on the maximum operating voltage and national or local regulations. Safety distances must be confirmed and signed by the tunnel operator.

Only splice boxes with grounding may be used for metal containing optical fibre sensor cables. Alternatively, metal-free sensor cables may be used without grounding.

1.5.3 Power Cable Monitoring



Warning!

Consider the danger of electrical voltage when working in the vicinity of power cables. They may carry life threatening voltage levels.

Only splice boxes with grounding may be used for metal containing optical fibre sensor cables. Alternatively, metal-free sensor cables may be used without grounding.

The grounding of metal containing optical fibre sensor cables must take into account induced currents and has to be compliant with the overall grounding concept of the power cable installation.



1.5.4 Explosive Areas



Warning!

The OTS3/LHD3-EX-series controllers are approved according to the European directive 94/9/EC and the IEC-Ex Scheme. In addition the LHD3-EX-series controllers are approved for fire detection by the VdS. The compliance to UL/ULC standards is not evaluated.

Only OTS3/LHD3-EX models of controllers equipped with suitable sensor cables may be used for temperature sensing or heat detection in explosive areas. The OTS3/LHD3-EX controller itself must be operated outside the explosive area or inside an approved EX –d enclosure. The user has to verify that the classification given in the approval documents of the equipment comply with the requirements of his application. Restrictions of using the OTS3/LHD3-EX-series controllers, protection boxes and the sensor cables in explosive given in the approval documents need to be considered.

We recommend operating the OTS3 / LHD3-EX-series controller choosing the option STOP MEASUREMENT as fibre break behaviour. For fire safety applications in explosive areas, the user has to choose the proper fibre break behaviour option based on a risk analysis of his application.

For the installation of sensor cables in explosive areas the requirements of national regulations for installation have to be considered (e.g. in Europe the standard EN 60079-14).

Only sensor cables, splice boxes and mounting components that are no source of ignition and do not cause other danger of explosion, e.g. by electrostatic discharge, must be used in explosive areas. Always ensure that no danger emerges from mechanical or other damage to the sensor cable. This applies to all production, installation, maintenance work and to regular operation in the object being monitored.

Conductive sensor cables must be protected against the ingress of energy from other sources (current sources, lightning, electrostatics, electromagnetic or magnetic fields) and its transmission into the explosive area. This requires a proper grounding, generally just outside the explosive area.

Splicing of fibres is forbidden in explosive areas. Ensure that no explosive agent or atmosphere is present during splicing of sensor cables.

National regulations must be considered during work in explosive areas.

The OTS3/LHD3-EX-series controller must be retested by LIOS Technology every 10 years in order to confirm the compliance to the explosion safety standards IEC / EN 60079-0 and -28.



LIOS FIBER OPTIC LINEAR **HEAT DETECTOR SOFTWARE**

Operations & Maintenance Manual
December 2015

LIOS Distributed Temperature Visualisation and Configuration Software Charon3



General Charon3 software allows easy configuration of LIOS distributed temperature sensing (DTS) systems and visualisation of temperature data. Charon3 guides the user during system commissioning and maintenance. Several languages and systems of units are supported. The measurement and configuration data are stored in a database for convenient access. Data visualisation and configuration reports present desired information in a comprehensive shape. The enhanced visualisation options of Charon3 offer various customisable views of data. Data export, import and software interfaces (ASCII, XML) support third party tools. Optional for EN.SURE systems real time thermal rating (RTTR) provides information about energy cable conductor temperature and prediction of critical operation conditions.



Technical Background Charon3 is a multithreaded 32 bit Windows application. State of the art object oriented software techniques are adopted using the Borland Delphi development environment. Measurement and configuration data is stored in a database using the Borland Database Engine and Paradox tables. A standard Windows installation and maintenance procedure can be used for installation, update and removal. Communication with the DTS system can be established over USB and TCP/IP connections. Data compression algorithms are applied for efficient communication and minimum use of data storage. Data integrity is guaranteed by CRC data check. The GUI (Graphical User Interface) of Charon3 is compliant with

modern Microsoft Windows standards in order to offer intuitive operation to the user. A concept of user levels with different access rights is used to protect sensitive settings. The display language can be changed easily during normal program operation. Further languages can be easily implemented using the LIOS Technology language tool.



Excellent Visualisation Capabilities The optional enhanced zone view of Charon3 enables visualization of zones from different DTS systems on a single screen. Zones can be placed in the standard view using the "Drag and Drop" feature. Data from external point sensors can be stored in the database and visualized in the enhanced view. Charon3 receives stores and visualizes data from MODBUS, DNP3 or IEC870 data sources. Details of enhanced views can be zoomed.



Database Measurement and configuration data is stored in a common database. Database size is minimized by compressing all data. Historic measurement data can be deleted automatically after a user-defined period. With this feature, measurements can be run for unlimited time without overloading the database. Measurements during alarms, fibre breaks, errors and other events can be excluded from automatic deletion.

A hard disk capacity of 100GB is sufficient for about one year of continuous measurement with five DTS systems. Appropriate configuration of the Charon3 automatic data base cleanup function

serves for an optimum utilisation of storage.



Efficient and Convenient Program Operation

Configuration data can be displayed in overview reports. Controller entries can be generated in the tree diagram from imported configuration or measurement data. All measurement data displayed can be easily managed using the measurement data explorer. Subsequently recorded measurement data is stored in so called "sessions".

Various data types of a DTS system (e.g. temperature curve, zone view) are always shown synchronously. A list of all events (e.g. alarms, fibre break, error messages) can be displayed for each fibre in the measurement data explorer.

Double-clicking on an event opens the corresponding zone views and the scroll bars for this measurement session are positioned at the correct time. The import and export of measurement and configuration data is simple as only one export file is generated during each occasion. ASCII export offers additional formatting options. The temperature history is displayed in real time.

Configuration and status reports can be generated automatically. Double-clicking on a fibre in the tree view opens the corresponding measurement data window. Double-clicking on a temperature history window moves the scroll bar of the linked profile windows to the selected time. The organisation of the windows menu has been optimised in order to provide the complete overview of all visible windows.



Typical Performance

Charon3 can manage various connected DTS systems simultaneously. A maximum of

five DTS systems can be connected to Charon3. Further DTS systems can be connected depending on the performance of the PC hardware and software. Ethernet interface allows for maximum data transfer capabilities especially when using

multiple DTS systems.



Communication

DTS systems with Ethernet interface can communicate with Charon3 via the TCP/IP protocol.

The communication via the USB interface offers Plug & Play functions. All entries in the message window are tagged with a time stamp.



Point Sensor Data

Point sensor measurement data can be stored and visualised by Charon3. Each particular

sensor needs an appropriate connection. The connections are provided by LIOS Technology. The MODBUS/TCP, DNP3 or IEC 60870-5-104 protocols are used for data import. Please contact our technical support for information on other SCADA protocols.



RTTR - Real Time Thermal Rating

For EN.SURE systems, LIOS provides an optional integrated real time thermal

rating (RTTR) evaluation via a well defined interface between Charon3 visualisation software and a market leading cable ampacity program. These two technologies are efficiently combined for forming a system capable of computing the future cable ampacity based on real-time temperature measurements. The actual maximum temperature reading of each configured cable section and the actual electrical current reading are evaluated for dynamic cable rating based on IEC standardised methods (mainly IEC 60287 and IEC 60853). When the geometry of the installation is not included in the IEC standards, the RTTR evaluation uses finite element methodologies to complement the IEC calculations. RTTR calculation results can be transmitted to a SCADA system using the IEC 60870-5-104 protocol.



Data Export

Measurement and configuration data can be exported from the data base. The complete information

will be stored to a file in order to archive

the data or in order to transfer the data to another Charon3 installation. The data base export contains compressed data in a proprietary format. The ASCII data export can be used to transfer the measurement data to third party software (e.g.

Microsoft Excel). The WITSML export plug-in can generate XML files which may be automatically transferred by the remote backup process over a SSL encrypted network connection to a remote PC system.

Technical Data

Operating Systems

- Windows 7, 32-bit and 64-bit (service pack 1 or higher)
- Windows XP, 32-bit (service pack 3 or higher)

Minimum Hardware Requirements

For configuration, commissioning and maintenance access only	<ul style="list-style-type: none"> ▪ PC with Intel Pentium 4 (1 GHz) processor or comparable ▪ 512 MB of RAM memory ▪ 250 MB of free hard disk capacity
--	--

Recommended Hardware Requirements

For configuration, commissioning, maintenance access and visualisation	<ul style="list-style-type: none"> ▪ PC with Intel Pentium 4 (2 GHz) processor or comparable ▪ 2 GB of RAM memory ▪ 250 MB of free hard disk capacity ▪ UPS (uninterruptible power source or battery backup) for graceful system shut down in case of an unexpected power disconnection ▪ 19" Screen Monitor
--	---

For configuration, commissioning, maintenance access visualisation and Real Time Temperature Rating

- Intel QuadCore PC with 2,8 GHz processor or comparable
- 2 GB of RAM memory
- 1 GB of free hard disk capacity
- UPS (uninterruptible power source or battery backup) for graceful system shut down in case of an unexpected power disconnection
- 19" Screen

Display Languages

English, German, Simplified Chinese, others upon request

Systems of Units

SI, Imperial units

Performance

Maximum number of DTS configurations	Unlimited
Maximum number of active DTS controllers	5 or more (depending on measurement settings and PC performance)

RTRR Hardware Requirements(for EN.SURE only)

EN.SURE RTRR (option)

Real Time Thermal Rating Interface	<ul style="list-style-type: none"> ▪ Open interface definition ▪ Supports several RTRR en-
------------------------------------	--

	gines
RTRR Input data	<ul style="list-style-type: none"> ▪ DTS temperature data ▪ Unlimited number of point sensor data (e.g. electrical current, soil humidity, wind speed, wind direction)
RTRR output data	<ul style="list-style-type: none"> ▪ Calculated conductor temperature ▪ Ampacity at target temperature, Load Factor, Cyclic Rating Factor M ▪ Calculated ambient temperature and thermal resistivity of the soil ▪ Results of emergency calculations

	minimum, maximum and average values, zone alarms, alarm locations, error messages and notes
USB	LIOS proprietary communication protocol
Ethernet interface	TCP/IP, Socket RJ45
Communication protocols (option)	MODBUS/TCP, DNP3 over LAN, IEC 60870-5-104, IEC 61850

Data Export

Data base export	<ul style="list-style-type: none"> ▪ Measurement data in MEX format ▪ Configuration data in CG3 format
ASCII	Writes measurement data to a text file.
Automatic MEX export (option)	Exports measurement data in regular terms for backup or data transfer in MEX format.
MODBUS TCP (option)	Exports live zone and event data using MODBUS/TCP
Remote backup	Client Server based data transfer application. Schedules transfer jobs for sending measurement files via a SSL encrypted network connection

Communication

Data transfer	Temperature profiles, backscattering and frequency raw data, zone
---------------	---

LIOS Technology GmbH
LINEAR OPTICAL SENSORS
Schanzenstrasse 39 Building D9-D13
51063 Köln/Cologne
Germany

Tel.: +49 221 998870, Fax: +49 221 99887150
info@lios-tech.com

Further information about our products and services can be obtained at our internet site

www.lios-tech.com

MEMBER OF THE NKT GROUP

This Page Left Intentionally Blank

Linear Optical Sensors



COMMUNICATION MANUAL PART V: XML

OTS3- / LHD3-Series Controllers



OTS3- / LHD3-Series Controllers
Communication Manual Part V: XML
Revision 1.7

Revision History:

Rev.	Date	Changes
1.0	14-02-2012	initial release
1.1	10-03-2012	documentation of new features (testing release)
1.2	27-03-2012	documentation of new features (testing release)
1.3	04-06-2012	documentation of new features (testing release)
1.4	21-08-2012	add changes for Firmware 1.0.7 (fibre break in loop mode, device status)
1.5	09-11-2012	add changes for Firmware 1.1.0 ($\Delta\alpha$ and OTDR data)
1.6	27-03-2013	add changes for Firmware 1.1.1 (progress information, Schema file)
1.7	23-06-2014	add changes for Firmware 1.1.2 (rate-of-change data, events extension)

Property of LIOS Technology GmbH, Köln.

This document and its associated distribution media contain proprietary information of LIOS Technology GmbH and may not be used, copied, or distributed in any form without the written permission of LIOS Technology GmbH.

Copies of this document may be made for internal use by the legally licensed owner.

Copyright © LIOS Technology GmbH 2012 – 2014. All rights reserved.

LIOS Technology GmbH
Schanzenstraße 39
Building D9–D13
D-51 063 Köln
Germany
Phone (+49) 221 99887 0
www.lios-tech.com
E-Mail: info@lios-tech.com



Contents

1	Introduction	5
2	Connection Characteristics	5
2.1	Physical and Data Link Layer	5
2.2	Network Layer	5
2.3	Transport Layer	5
2.3.1	TCP keep-alive	6
2.4	Presentation Layer	6
2.4.1	Data compression	7
2.5	LIOS Application Layer Packets	7
3	Data buffer modes	7
4	LIOS XML Protocol Structure	8
4.1	Time stamps	9
4.2	Fibre measurement data	9
4.2.1	Fibre break positions	9
4.2.2	Fibre adres points	12
4.2.3	Temperature profile	12
4.2.4	Backscatter profile	12
4.2.5	Attenuation profile	13
4.2.6	OTDR profile	13
4.2.7	Rate-of-Change profile	14
4.2.8	Zone temperatures	14
4.2.9	Alarm triggering locations	14
4.3	System status data	15
4.3.1	System address points	15
4.3.2	Last event code	16
4.3.3	Device inputs and outputs	16
4.3.4	Sensor readings	16
4.4	Device status information	17
4.4.1	Device inputs and outputs	17
4.4.2	Measurement status	17
4.4.3	Controller statistical data	19
4.4.4	Fibre commissioning status	19
4.4.5	CF card status	19
4.5	Schema file	20
4.6	Measurement Progress	20
4.7	Command transmission	20
4.8	Command result	21



5	Commands	21
5.1	Reset Alarms	21
5.2	Acknowledgement	21
5.3	Device status	21
5.4	Start Measurement	22
5.5	Stop Measurement	22
5.6	Set Clock	22
5.7	Read Schema	22
6	Plug-In Configuration	23
A	LIOS XML Schema	25
B	Examples	35
B.1	Fibre measurement data	35
B.2	Device status information data	37

List of Figures

1	Structure of the top-level XML container	8
2	Structure of the fibre measurement data XML container	10
3	Structure of the system status data XML container	15
4	Structure of the device status information XML container	18
5	Activation of the XML Data and Control plug-in	23
6	XML Data and Control plug-in configuration	24

List of Tables

1	Address point condition flags	11
2	Alarm and pre-alarm source flags	11
3	Event codes	16



1 Introduction

The XML Data and Control Protocol option for the OTS3-/LHD3-Series Controllers provides inter alia status, alarm, zone, temperature and backscatter profile data of the system and all channels through an easy-to-use XML protocol over standard Ethernet connections. Additionally, it offers a set of configuration and control options as to modify the behaviour of the system.

Note: In this document, the keywords *MUST*, *MUST NOT*, *REQUIRED*, *SHOULD*, *SHOULD NOT* and *MAY* are to be interpreted as described in RFC 2119¹.

Some features are available in newer firmware versions only. They are marked with **Version** at **x.x.x.x** the margin.

2 Connection Characteristics

The low-level characteristics according to layers 1 up to 7, inclusively, of the OSI model are presented in this section.

Note: The session layer (OSI layer 5) is not used.

2.1 Physical and Data Link Layer

The XML Data and Control Protocol is available through the Ethernet interfaces of the OTS controller.

Note: Using this protocol over the USB or EIA/TIA232 interfaces of the OTS controller is *not* supported.

The characteristics of the physical and data link layer (OSI layers 1 and 2) of the connection are configured globally and used for all protocols. The system supports auto-negotiation or fixed settings according to the standards IEEE 802.3 Clause 14 and Clause 24 (10 MBit/sec and 100 MBit/sec, half- or full-duplex, respectively).

2.2 Network Layer

The Internet Protocol (IPv4) according to RFC 791² is used as network layer (OSI layer 3). The IP address of the controller is configured globally and used for all protocols.

2.3 Transport Layer

The transport layer (OSI layer 4) uses the Transmission Control Protocol (TCP) according to RFC 793³. The OTS controller operates as server, the remote system as client, i.e. the remote system *MUST* use the IEEE 1003.1-2008 `connect ()` system call (or the equivalent on systems which do not comply with the standards) to establish the connection.

The TCP port where the OTS controller listens for incoming connections is configurable and defaults to port 9974.

Up to eight simultaneous connections are supported if data buffering on the CF card is disabled. All connected peers will receive the same data. If data buffering is enabled, only one simultaneous connection is allowed (see section 3 on page 7 for details).

¹See <http://tools.ietf.org/html/rfc2119>

²See <http://tools.ietf.org/html/rfc791>

³See <http://tools.ietf.org/html/rfc793>



2.3.1 TCP keep-alive

As to detect broken connections, the OTS controller will always enable TCP-level keep-alive probes according to RFC 1122⁴. Thus, the remote system *MUST* reply with properly formatted packets. Otherwise, the connection will automatically be terminated by the OTS controller when the keep-alive time expires.

The remote system *SHOULD* also enable sending keep-alive probes, typically by using the IEEE 1003.1-2008 `setsockopt()` system call with the `SO_KEEPALIVE` socket option.

2.4 Presentation Layer

On the presentation layer (OSI layer 6), the OTS controller always uses authentication and encryption according to the Transport Layer Security (TLS) protocol Version 1.0⁵. During the negotiation phase, the OTS controller will present a self-signed X.509 certificate with a well-known fingerprint and Common Name (CN):

- CN: LIOS_OTS3/LHD3_Series/XML_Data_and_Control
- SHA1 fingerprint: 76:2D:40:FE:09:DE:95:AE:AA:11:6C:F1:7A:1B:42:A4:A6:D2:D7:7C

Future extension: The default certificate *MAY* be replaced by a different certificate and associated private key. The private key *MUST NOT* be encrypted. Both the certificate and the private key *MUST* be contained in a PEM (base64 encoded ASCII) file.

Future extension: The remote system *MAY* also send a certificate to the OTS controller. In this case, the OTS controller *MUST* be configured to either check the certificate against a CA certificate which has been uploaded to the controller as PEM (base64 encoded ASCII) file, or to check the certificate's fingerprint against a configured value.

The user may switch off TLS completely, i.e. use a plain TCP connection without any security and compression. Note that this operation mode is *not* recommended.

The TLS implementation in the OTS controller will accept only a limited set of "high grade" ciphers:

- Diffie-Hellman (DH) key exchange with RSA or DSS authentication
- RSA key exchange with RSA authentication
- 128-bit or 256-bit AES encryption
- SHA1 Message Digest (MAC)

The remote system *MUST* support at least one of the aforementioned combinations. Note that in particular all "weak" ciphers cannot be used.

Using the Open Source *OpenSSL*⁶ software, the connection can easily be tested by running the command

```
openssl s_client -tls1 -connect <OTS IP address or host name>:9974
```

which will print the connection parameters and then dump all received data.

⁴See <http://tools.ietf.org/html/rfc1122>

⁵See <http://tools.ietf.org/html/rfc2246>

⁶See <http://openssl.org/>



2.4.1 Data compression

The OTS controller will always offer to use the “deflate” TLS compression method according to RFC 3749⁷ as to reduce the bandwidth and latency requirements of the connection. The remote system *SHOULD* also support this option. If this is not possible, the connection will fall back to uncompressed mode (“CompressionMethod.null”).

2.5 LIOS Application Layer Packets

The data exchange is based on *packets* (application layer, OSI layer 7), each containing exactly one complete XML Version 1.0 compliant document⁸. As to simplify the exchange, the XML document is transmitted in the following form:

- size of the XML document in bytes as 32-bit unsigned integer value in little-endian byte order
- XML document as UTF-8 encoded character data which *MUST NOT* be ASCII NUL-terminated.

This can be represented as the following C structure:

```
#include <inttypes.h>
struct LIOSXmlPacketT {
    uint32_t xmlCharCount;
    char xmlData[0]; // place-holder for the XML data string
};
```

Thanks to the size information preceding the real data, the receiving instance can allocate a structure of the appropriate size, map it to the data type above, and call `SSLread()`⁹ to receive the complete XML document.

3 Data buffer modes

The XML Data and Control Protocol plug-in can be configured to optionally store data frames on the built-in CF card of the controller if no peer connection is open in order to prevent the loss of data. 1.0.7.0

Note The data buffering mode is supported since Firmware 1.0.7.

The data buffer can operate either in *ring buffer mode* or in *linear buffer mode*. For both modes, the maximum fill level of the CF card is 50% to ensure that sufficient space for event and alarm recorder data is kept free. In ring buffer mode, the oldest buffered files will be erased when the maximum fill level is reached, whereas in linear buffer mode storing files will stop in this case.

The buffered data is transmitted automatically when the peer opens the connection. Note that the buffered data and any new frames may be “mixed” during this phase, i.e. the receiving peer must check the time stamps. All buffered data blocks which were transmitted are erased from the CF card.

Note Only *one* connection may be open simultaneously when the data buffer mode is enabled. If the data buffer mode is disabled, up to *eight* connections may be open simultaneously. In the latter case, the user may use multiple connections to implement a redundant infrastructure.

⁷See <http://tools.ietf.org/html/rfc3749>

⁸See *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, W3C Recommendation 26 November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126/>.

⁹When using the `openssl` library.



4 LIOS XML Protocol Structure

The OTS controller will send XML data packets and receive commands and configuration changes in XML data packets which must be valid according to the LIOS Technology XML Protocol Schema file (see appendix A on page 25). Every XML data packet sent by the OTS controller starts with the UTF-8 *Byte Order Mark* character¹⁰.

The top-level container `lios:LIOSXml` contains exactly *one* of the following elements (see figure 1):

- fibre measurement data `measData` (see section 4.2 on the next page). This element is sent only from the OTS controller to the remote system. The remote system *MUST NOT* send such an element to the controller;
- system status data `systemData` (see section 4.3 on page 15). This element is sent only from the OTS controller to the remote system. The remote system *MUST NOT* send such an element to the controller;
- status information data `statusData` (see section 4.4 on page 17). This element is sent only from the OTS controller to the remote system as reply to a `status` command (see section 5.3 on page 21). The remote system *MUST NOT* send such an element to the controller;
- a single `command` (see section 4.7 on page 20). This element is sent only from the remote system to the OTS controller;
- measurement progress information `progress` (see section 4.6 on page 20). This element is sent only from the OTS controller to the remote system. The remote system *MUST NOT* send such an element to the controller;
- the result of executing a command `result` (see section 4.8 on page 21). This element is sent only from the OTS controller to the remote system. The remote system *MUST NOT* send such an element to the controller;

¹⁰Zero width non-breaking space, UTF-8 character 0xef 0xbb 0xbf.

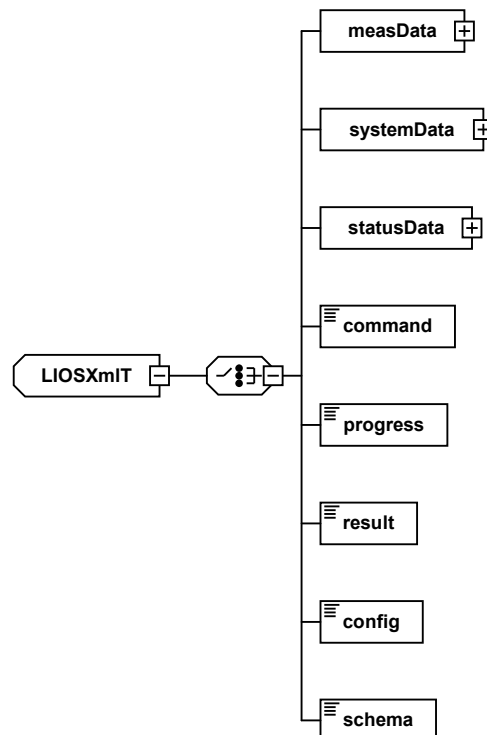


Figure 1: Structure of the top-level XML container



- the LIOS XML Schema file `schema` (see section 4.5 on page 20). This element is sent only from the OTS controller to the remote system as reply to a `schema` command (see section 5.3 on page 21). The remote system *MUST NOT* send such an element to the controller.

4.1 Time stamps

Time stamps are transmitted in `xsd:dateTime` format with a `xsd:boolean` NTP synchronisation status. The device will *always* report the time in *Universal Time* (UT; "Zulu"):

```
<timeStamp ntp="1">2011-10-26T14:17:37.567Z</timeStamp>
```

4.2 Fibre measurement data

Measurement data is sent automatically by the OTS controller as soon as it is available. The `measData` XML container has five attributes:

- `fibre` contains the channel number which produced the measurement, counting from 0 for the first channel;
- `fibreName` contains the user-defined name of the fibre;
- `otsModel` contains the model description of the OTS controller;
- `otsSerial` contains the serial number of the OTS controller;
- `otsName` contains the user-defined name of the OTS controller.

The amount of data which is transmitted in the container (see figure 2 on the following page) can be configured. Note that the time stamp (see section 4.1) cannot be disabled. If none of the aforementioned measured values is enabled for transmission, the device will not send any fibre measurement data.

An example fibre measurement data file is included in appendix B.1 on page 35.

4.2.1 Fibre break positions

This value is transmitted *only* if a fibre break has been detected. The format of the transmitted data depends upon the configuration of the fibre:

- in *single-ended mode*, the `breakPos` container contains the fibre break position in meter relative to the fibre reference point as `xsd:float`:

```
<breakPos>891</breakPos>
```

- in *loop mode*, a `breakRange` container containing the fibre break area as start and end position in meter relative to the fibre reference point as two `xsd:float` values is transmitted: 1.0.7.0

```
<breakRange>
  <start>2576</start>
  <end>3184</end>
</breakRange>
```

Note Loop mode is supported since Firmware 1.0.7.

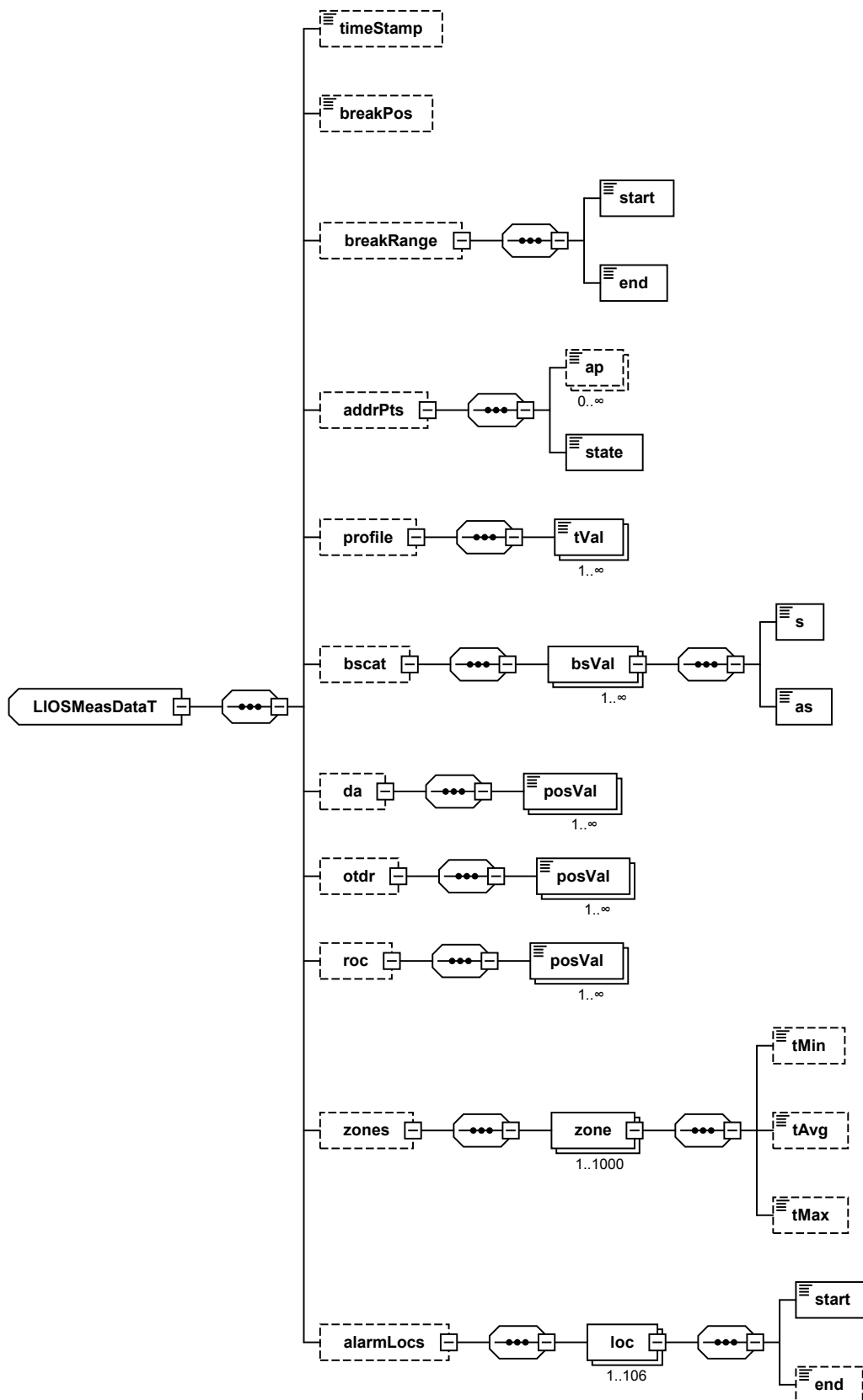


Figure 2: Structure of the fibre measurement data XML container



Flags	Description
01xx _{hex}	The address point indicates a fibre alarm condition. The lower 8 bits xx contain a bit mask indicating the reasons triggering the alarm (see table 2).
0201 _{hex}	The address point indicates a fibre break condition. The lower 8 bits always contain the value 1.
03xx _{hex}	The address point indicates a fibre pre-alarm condition. The lower 8 bits xx contain a bit mask indicating the reasons triggering the pre-alarm (see table 2)
0400 _{hex} , 0401 _{hex}	The address point indicates a control function change. The lower 8 bits contain the value 1 if the control function was activated, an 0 if it was deactivated.
0501 _{hex}	The address point is related to a fire size event. The lower 8 bits always contain the value 1.
0601 _{hex}	The address point is related to a fire direction event. The lower 8 bits always contain the value 1.
0701 _{hex}	The address point indicates the activation of an alarm input. The lower 8 bits always contain the value 1.
0800 _{hex} , 0801 _{hex}	The address point indicates the change of a control input. The lower 8 bits contain the value 1 if the input was activated, an 0 if it was deactivated.
0901 _{hex}	The address point indicates an acknowledgement operation. The lower 8 bits always contain the value 1.
0a01 _{hex}	The address point indicates a reset operation. The lower 8 bits always contain the value 1.
0b01 _{hex}	The address point indicates that the test mode has been started. The lower 8 bits always contain the value 1.
0c01 _{hex}	The address point indicates that the test mode has been stopped. The lower 8 bits always contain the value 1.
0d01 _{hex}	The address point is related to an error event. The lower 8 bits always contain the value 1.
0e01 _{hex}	The address point is related to a warning event. The lower 8 bits always contain the value 1.
0f00 _{hex} , 0f01 _{hex}	The address point indicates the change of the <i>mute buzzer</i> control input. The lower 8 bits contain the value 1 if the input was activated, an 0 if it was deactivated.

Table 1: Address point condition flags. The higher 8 bits of the 16-bit flags value indicates the triggering condition, the lower 8 bits contain further information according to the triggering condition.

Bit	Mask	Description
0	01 _{hex}	Indicates that the maximum alarm criterion triggered the alarm or pre-alarm.
1	02 _{hex}	Indicates that the minimum alarm criterion triggered the alarm or pre-alarm.
2	04 _{hex}	Indicates that the <i>hot spot</i> alarm criterion triggered the alarm or pre-alarm.
3	08 _{hex}	Indicates that the <i>cold spot</i> alarm criterion triggered the alarm or pre-alarm.
4	10 _{hex}	Indicates that the first differential alarm criterion triggered the alarm or pre-alarm.
5	20 _{hex}	Indicates that the second differential alarm criterion triggered the alarm or pre-alarm.
6	40 _{hex}	Indicates that the third differential alarm criterion triggered the alarm or pre-alarm.
7	80 _{hex}	Indicates that the alarm or pre-alarm has been triggered while the system is in test mode.

Table 2: Alarm and pre-alarm source flags. If more than one criterion is active, the flags contain the logical OR of all triggered criteria.



4.2.2 Fibre address points

The fibre address point states are transmitted as a list of address point numbers with a `flags` attribute each (both `xsd:unsignedShort`) giving more information about the address point (see [table 1 on the previous page](#)).

The `state` element contains a `xsd:hexBinary`¹¹ encoded bit mask of the trigger states of all 3100 fibre address points in their “natural” order, i.e. the first two hex digits contain the states of address points 3096 to 3099, inclusively, and 4 unused bits which are always 0, the second two hex digits the states of address points 3088 to 3095, inclusively, etc. The last two hex digits contain the states of address points 0 to 7, inclusively:

```
<addrPts>
  <ap flags="257">2</ap>
  <ap flags="257">4</ap>
  <ap flags="769">1007</ap>
  <ap flags="257">8</ap>
  <state>0000000[...]</state> <!-- 776 hex digits -->
</addrPts>
```

4.2.3 Temperature profile

Each value gives the temperature in °C at a position in meter relative to the fibre reference point which is included in the `pos` attribute of the value (both are `xsd:float` values). The `count` attribute of the `profile` container (type `xsd:nonNegativeInteger`) indicates the number of sampling points in the temperature profile. The `tRef` attribute (type `xsd:float`) gives the temperature of the internal reference coil. Note that illegal temperature values or data behind a fibre break is *not* transmitted, i.e. the number of `tVal` elements may be smaller than the count:

```
<profile count="6315" tRef="28.0316">
  <tVal pos="0">26.5565</tVal>
  <tVal pos="1">25.4472</tVal>
  <tVal pos="2">24.6831</tVal>
  <tVal pos="3">23.4052</tVal>
  <!-- more values -->
</profile>
```

4.2.4 Backscatter profile

The values for each spatial point are given in `bsVal` containers, of which the `s` child gives the Stokes and the `as` child the Antistokes values, respectively. The `pos` attribute gives the position relative to the fibre reference point. Note that the *complete* backscatter profile, including the part inside the controller, will be transmitted. Thus, the backscatter profile will *always* start at negative positions. The `count` attribute of the `bscat` container (type `xsd:nonNegativeInteger`) indicates the number of sampling points in the backscatter profile. All values are `xsd:float`:

```
<bscat count="4896">
  <bsVal pos="-126">
    <s>29115.7</s>
    <as>29095</as>
  </bsVal>
  <bsVal pos="-123">
```

¹¹Note that the definition of this data type requires an even number of hex digits.



```
<s>38122.4</s>
<as>28746.8</as>
</bsVal>
<!-- more values -->
</bscat>
```

Note Backscatter profiles are transmitted only if the *Transmit Backscattering Profile* option is enabled in the Measurement Parameters of the respective fibre. See the *LHD3-/OTS3-Series Controllers Operation Manual* for details.

4.2.5 Attenuation profile

Each `posVal` element gives the value at a position in meter relative to the fibre reference point which is included in the `pos` attribute of the value (both are `xsd:float` values). The `count` attribute of the `da` container (type `xsd:nonNegativeInteger`) indicates the number of sampling points in the profile:

1.1.0.0

```
<da count="1360">
  <posVal pos="-126">0</posVal>
  <posVal pos="-123">0</posVal>
  <!-- all values in the device are 0 -->
  <posVal pos="18">-8.09443e-06</posVal>
  <posVal pos="21">0.000164516</posVal>
  <posVal pos="24">-0.00037823</posVal>
  <!-- more values -->
</da>
```

Note Attenuation data along the fibre ($\Delta\alpha$ profile) is transmitted only in loop mode and only if the *Transmit Backscattering Profile* option is enabled in the Measurement Parameters of the respective fibre. See the *LHD3-/OTS3-Series Controllers Operation Manual* for details.

Attenuation data along the fibre ($\Delta\alpha$ profile) is available since firmware version 1.1.0. It is not available from *Linear Heat Detection* (LHD) systems.

4.2.6 OTDR profile

The `posVal` elements are to be interpreted as for the z-dependent attenuation data (see above):

1.1.0.0

```
<otdr count="1318">
  <posVal pos="-126">22135.5</posVal>
  <posVal pos="-123">22180.7</posVal>
  <posVal pos="-120">22200.6</posVal>
  <posVal pos="-117">22190</posVal>
  <posVal pos="-114">22170.8</posVal>
  <!-- more values -->
</otdr>
```

Note OTDR profile data is transmitted only if the *Transmit Backscattering Profile* option is enabled in the Measurement Parameters of the respective fibre. See the *LHD3-/OTS3-Series Controllers Operation Manual* for details.

OTDR profile data is available since firmware version 1.1.0. It is not available from *Linear Heat Detection* (LHD) systems.



4.2.7 Rate-of-Change profile

1.1.2.0 Each value gives the temperature rate-of-change in °C/second at a position in meter relative to the fibre reference point which is included in the `pos` attribute of the value (both are `xsd:float` values). The `count` attribute of the `roc` container (type `xsd:nonNegativeInteger`) indicates the number of sampling points in the rate-of-change profile. Note that data behind a fibre break is *not* transmitted, i.e. the number of `posVal` elements may be smaller than the count:

```
<roc count="6315">
  <posVal pos="0">0.0066704</posVal>
  <posVal pos="0.5">-0.00306263</posVal>
  <posVal pos="1">0.018405</posVal>
  <posVal pos="1.5">0.0293577</posVal>
  <posVal pos="2">0.00996308</posVal>
  <posVal pos="2.5">0.0135454</posVal>
  <!-- more values -->
</roc>
```

Note Rate-of-change data is available since firmware version 1.1.2.

4.2.8 Zone temperatures

This block includes zone minimum and maximum temperatures and positions, and zone average temperatures. All temperatures are given in °C. The minimum and maximum temperatures include the position in meter relative to the fibre reference point where the value has been detected. The `zone` container includes the attribute `idx` which gives the absolute zone index, starting with 0:

```
<zones>
  <zone idx="0">
    <tMin pos="551">20.7216</tMin>
    <tAvg>23.8281</tAvg>
    <tMax pos="248">26.7803</tMax>
  </zone>
  <!-- more zones -->
</zones>
```

4.2.9 Alarm triggering locations

Alarm triggering locations can be transmitted as single trigger positions or as triggered regions, both in meter relative to the fibre reference point.

A triggered region is given as `start` and `end` values in a `loc` container, whereas a single (isolated) trigger position is indicated by a `loc` container containing only the `start` value:

```
<alarmLocs>
  <loc>
    <start>630</start>
    <end>1260</end></loc>
  <loc>
    <start>1890</start>
  </loc>
</alarmLocs>
```

4.3 System status data

System status data is sent automatically by the OTS controller whenever one of the states changes. The `systemData` XML container has three attributes:

- `otsModel` contains the model description of the OTS controller;
- `otsSerial` contains the serial number of the OTS controller;
- `otsName` contains the user-defined name of the OTS controller.

The amount of data which is transmitted in the container (see figure 3) can be configured. Note that the time stamp (see section 4.1 on page 9) cannot be disabled. If none of the aforementioned values is enabled for transmission, the device will not send any system status data.

4.3.1 System address points

System address points are transmitted in the same format as fibre address points (see section 4.2.2 on page 12 for details). The `state` element contains a `xsd:hexBinary` encoded bit mask of the trigger states of all 256 system address points, i.e. 64 hex digits.

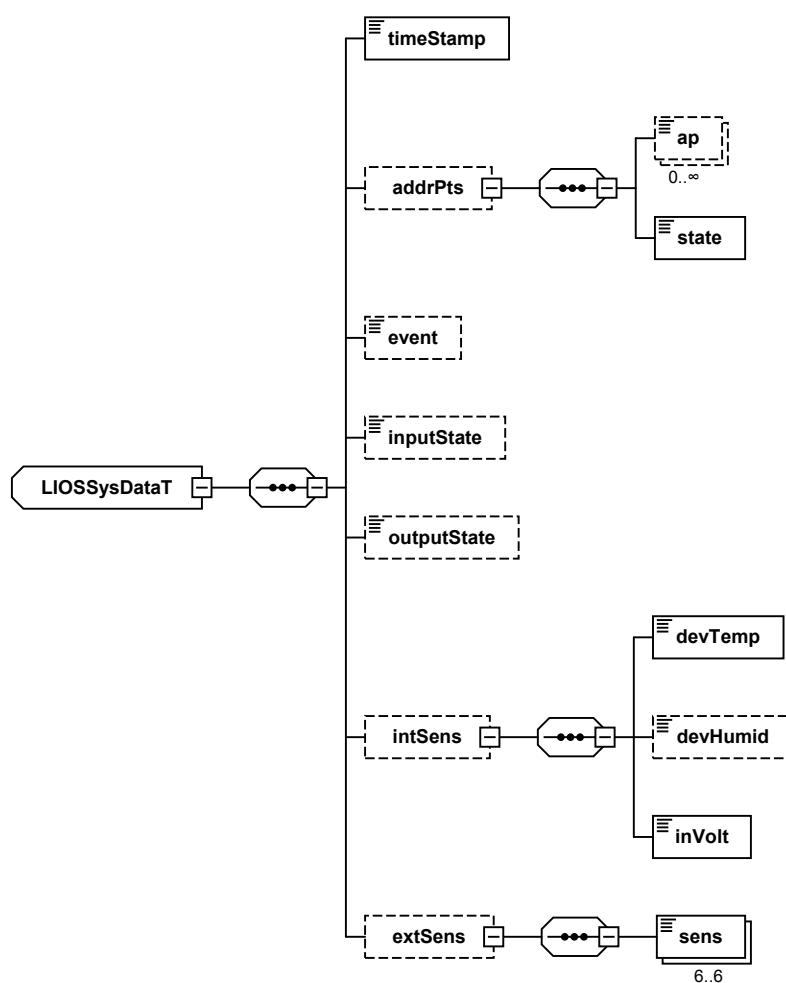


Figure 3: Structure of the system status data XML container



4.3.2 Last event code

The last event code is transmitted as a `xsd:nonNegativeInteger` number. See table 3 for a list of event codes:

```
<event>107</event>
```

4.3.3 Device inputs and outputs

The `inputState` and `outputState` containers contain a bit mask of the logical states¹² of the programmable device inputs and outputs, respectively, encoded as `xsd:hexBinary`. The states are given in their “natural” order, i.e. the first two hex digits contain the states of the highest-order inputs or outputs, while the last two hex digits contain the states of the lowest-order inputs or outputs.

The `sysFault` and `sysAlarm` attributes (both `xsd:boolean`) of the `outputState` container give the logical state of the common fault and common alarm outputs, respectively:

```
<inputState>0000000008</inputState>
<outputState sysFault="0"
  sysAlarm="1">0000000000000000000000000010</outputState>
```

4.3.4 Sensor readings

Readings of the internal sensors include the device temperature in °C, the internal device humidity in % relative humidity, and power supply input voltage in VDC. Note that the humidity is reported as `NaN` if the device is not equipped with a humidity sensor:

¹²Note that the programmable outputs and inputs may individually be configured as active high or active low. The common fault output is closed if no fault has been detected, the common alarm output is closed if any alarm occurred.

Code	Description
100	Configuration Error: Configuration is not loaded by the controller.
101	Fibre break: Unexpected change of the sensor cable length.
102	Measurement error: Invalid measurement.
103	Device error, call support: Controller hardware error requiring support.
104	Device error: Controller error, e.g. low buffer battery.
105	Environmental conditions error: Supply voltage, temperature and/or humidity are out of range. Measurement is stopped.
106	Handling error: No fibre, ...
107	Measurement stopped: Message sent once when the measurement is stopped.
108	Measurement inactive: Periodic message when the measurement is inactive.
109	Alarm zones deactivated
200	Configuration warning: Configuration data is not valid.
201	Measurement warning: Single invalid measurement. This measurement is disregarded, and measurements are continued.
202	Environmental conditions warning: Supply voltage, temperature and/or humidity are close to the limits.
203	Communication warning: Low quality of data transmission.

Table 3: Event codes reported in the system data `event` container



```
<intSens>
  <devTemp>38</devTemp>
  <devHumid>NaN</devHumid>
  <inVolt>23.923</inVolt>
</intSens>
```

If the device is equipped with an optional analogue sensors add-on card, their readings are given as 6 `xsd:float` values:

```
<extSens>
  <sens idx="0">38.6</sens>
  <sens idx="2">0.413</sens>
  [...] <!-- 6 values -->
</extSens>
```

4.4 Device status information

The device status information is transmitted as reply to the status command (see section 5.3 on page 21). 1.0.7.0
The `statusData` XML container has three attributes:

- `otsModel` contains the model description of the OTS controller;
- `otsSerial` contains the serial number of the OTS controller;
- `otsName` contains the user-defined name of the OTS controller.

The container always includes the time stamp (see section 4.1 on page 9). The following sections describe the additional data (see figure 4). An example device status information data file is included in appendix B.2 on page 37.

Note The status command and thus the `statusData` XML container are supported since firmware 1.0.7.

4.4.1 Device inputs and outputs

The states of the device inputs and outputs is transmitted in the `inputState` and `outputState` containers (see section 4.3.3 on the preceding page for details).

4.4.2 Measurement status

The measurement status is transmitted in the `measState` container, giving the currently measuring fibre index, starting with 0 for the first channel, in the `activeFibre` (type `xsd:int`), the test mode status in the `testMode` (type `xsd:boolean`), and the alarm processing status in the `alarmProcessing` (type `xsd:boolean`) containers, respectively.

If no regular measurement is active, the value for the measuring fibre is -1. The alarm processing state is TRUE if the alarm processing configuration is valid, and if all measurement channels in the measurement sequence have a valid zone configuration, i.e. they can produce alarm, pre-alarm or fibre break address points:

```
<measState>
  <activeFibre>3</activeFibre>
  <testMode>0</testMode>
  <alarmProcessing>1</alarmProcessing>
</measState>
```

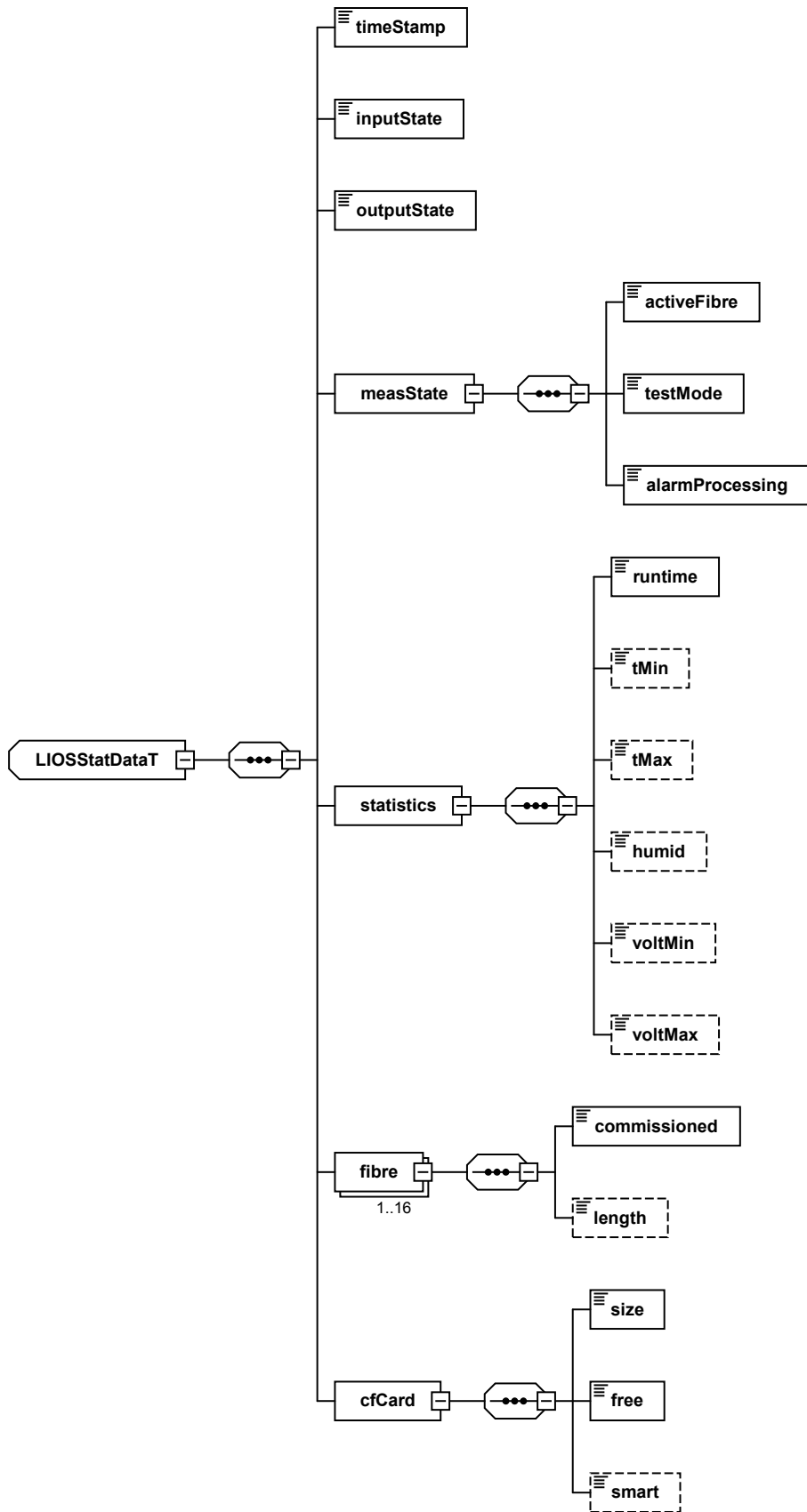


Figure 4: Structure of the device status information XML container



4.4.3 Controller statistical data

Statistical data of the controller is transmitted in the `statistics` container. The container includes the following elements, collected since the last reset of the user statistics:

- the run time in seconds in the `runtime` element (type `xsd:nonNegativeInteger`);
- the minimum and maximum internal device temperature in °C as `xsd:float` in the `tMin` and `tMax` elements, respectively. The `time` attribute gives the time when the value has been recorded;
- the maximum internal relative humidity in % as `xsd:float` in the `humid` element. The `time` attribute gives the time when the value has been recorded. If the controller is not equipped with an internal humidity sensor, the value is always NaN;
- the minimum and maximum DC input voltage of the power supply in V DC as `xsd:float` in the `voltMin` and `voltMax` elements, respectively. The `time` attribute gives the time when the value has been recorded.

```
<statistics>
  <runtime>5835262</runtime>
  <tMin time="2012-03-06T08:44:56">20</tMin>
  <tMax time="2012-05-24T14:10:46">42</tMax>
  <humid time="2012-06-06T11:42:06">35.7</humid>
  <voltMin time="2012-08-15T08:30:45">23.6918</voltMin>
  <voltMax time="2012-03-15T09:12:20">24.0578</voltMax>
</statistics>
```

Note that fields are omitted after a reset of the user statistics when no new data has yet been collected (during about 30 seconds after the reset of the user statistics).

4.4.4 Fibre commissioning status

The commissioning state of all fibres is included in a series of `fibre` containers, with attributes `index` giving the absolute fibre index, starting with 0, as `xsd:nonNegativeInteger` and the fibre name in `fibreName` as `xsd:string`. Each container includes a `xsd:boolean` element `commissioned` indicating if the fibre has been commissioned. The optional element `length` is present only for commissioned fibres and gives as `xsd:float` the detected fibre length in m:

```
<fibre index="1" fibreName="Tunnel Tube 2">
  <commissioned>1</commissioned>
  <length>508</length>
</fibre>
```

4.4.5 CF card status

The status of the built-in CF memory card is transmitted in in the `cfCard` container. The `size` and `free` elements (both `xsd:nonNegativeInteger`) are always present and give the total size and the free space on the CF card, both in kByte, respectively. The `xsd:boolean` element `smart` is present only if the CF card supports *Self-Monitoring, Analysis and Reporting Technology* (S.M.A.R.T.). The value is TRUE if a “good” S.M.A.R.T. has been detected:

```
<cfCard>
  <size>1995936</size>  <!-- a 2 GByte CF card -->
  <free>1798120</free>
  <smart>1</smart>
</cfCard>
```




4.5 Schema file

- 1.1.1.0** The LIOS XML Schema file is transmitted as reply to the schema command (see section 5.7 on page 22). The returned `schema` XML container contains the full Schema file including the annotations as used by the firmware. The file is deflated according to RFC 1950¹³ and then transmitted in `xsd:base64Binary` format.

Note Reading the schema file from the device is supported since firmware 1.1.1.

4.6 Measurement Progress

- 1.1.1.0** Measurement progress information is sent automatically approximately every 5 seconds by the OTS controller. The `progress` XML container has five attributes:

- `fibre` contains the channel number which produced the measurement, counting from 0 for the first channel;
- `fibreName` contains the user-defined name of the fibre;
- `otsModel` contains the model description of the OTS controller;
- `otsSerial` contains the serial number of the OTS controller;
- `otsName` contains the user-defined name of the OTS controller.

The contents of the container is the measurement progress, given in %, as `xsd:float` value.

Note Measurement progress data will never be stored on the built-in CF card.
The measurement progress is supported since firmware 1.1.1.

4.7 Command transmission

A command on the OTS controller is executed by sending a XML packet with *MUST* contain exactly one `command` container. The command name is transmitted as contents of the container. If the command has an argument, it is added as attribute `arg` to the `command` container. Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<lios:LIOSXml xmlns:lios="http://www.lios-support.com/LIOSXml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://www.lios-support.com/LIOSXml LIOSXml.xsd"
  version="1.8">
  <command arg="3">start</command>
</lios:LIOSXml>
```

Note All attributes of the `lios:LIOSXml` are optional and *MAY* be omitted.

See section 5 on the next page for a list of all commands supported by the OTS controller. The OTS controller will reply immediately with a `result` indicating whether the operation has been successful or not (see section 4.8 on the facing page for details).

¹³P. DEUTSCH and J-L. GAILLY: *ZLIB Compressed Data Format Specification version 3.3* (<http://www.ietf.org/rfc/rfc1950.txt>). With the Open Source "ZLib" library, available from <http://www.zlib.net/>, the data can be inflated using the `uncompress()` or the `inflateInit()`, `inflate()` and `inflateEnd()` methods.



4.8 Command result

After receiving a command or a configuration, the OTS controller will always reply with a `result` container with a `xsd:boolean` attribute indicating whether the operation has been successful (value `true`) or not (value `false`). In the latter case, the contents of the `result` container is a human-readable message describing why the command execution failed. On success, the `result` container is empty:

```
<result ok="1"/>
```

5 Commands

This section gives an overview over all commands which can be sent to the OTS controller.

Notes Some commands cannot be used if the controller has been switched to nonreactive mode. See the *LHD3-/OTS3-Series Controllers Operation Manual* for details about the nonreactive mode of the OTS controller.

For some commands, the *OTS Remote Control over XML* license is needed. Contact LIOS Technology or your support partner for details.

5.1 Reset Alarms

Command: `reset`

Argument: none

This command performs the *Reset* operation of the OTS controller. See the *LHD3-/OTS3-Series Controllers Operation Manual* for details. It cannot be used if the controller has been switched to nonreactive mode.

5.2 Acknowledgement

Command: `ack`

Argument: none

This command performs the *Acknowledgement* operation of the OTS controller. See the *LHD3-/OTS3-Series Controllers Operation Manual* for details. It cannot be used if the controller has been switched to nonreactive mode.

5.3 Device status

Command: `status`

1.0.7.0

Argument: none

The command requests the current device status which is reported as a `statusData` container (see section 4.4 on page 17 for details).

Note This command is supported since Firmware 1.0.7.



5.4 Start Measurement

Command: `start`

Argument: optional: logical fibre number, counting from zero

If no argument is given, start the programmed measurement sequence. Otherwise, start a single-fibre measurement of the passed fibre. The *OTS Remote Control over XML* license is needed if the command shall be used. It cannot be used if the controller has been switched to nonreactive mode.

Note The logical fibre numbers are counted from zero.

5.5 Stop Measurement

Command: `stop`

Argument: none

Stop the measurement. The *OTS Remote Control over XML* license is needed if the command shall be used. It cannot be used if the controller has been switched to nonreactive mode.

5.6 Set Clock

1.1.0.0 **Command:** `time`

Argument: new UT time in XML format (`xsd:dateTime`)

Set the Real Time Clock (RTC) of the controller to the passed time. The *OTS Remote Control over XML* license is needed if the command shall be used. Note that any fractional part of the seconds will be ignored. The time *must* be given as UT (Zulu time zone "Z"). The command will fail if NTP is activated and the controller is synchronised with a NTP server.

Note This command is supported since Firmware 1.1.0.

5.7 Read Schema

1.1.1.0 **Command:** `schema`

Argument: none

The command requests the LIOS XML Schema file `schema` container (see section 4.5 on page 20 for details).

Note Reading the schema file from the device is supported since firmware 1.1.1.

6 Plug-In Configuration

In order to use the XML Data and Control plug-in, it first has to be activated and configured (see figure 5):

1. Select the OTS controller in the tree view
2. Activate the *Device* tab.
3. In the *Controller* group, click *Communication*.
4. Select *Other Protocols* from the menu.
5. in the *Other Protocols* dialogue, activate the XML Data and Control plug-in using the check box.
6. Click *Configure* to configure the XML Data and Control plug-in options.

In the XML Data and Control plug-in options dialogue (see figure 6 on the following page), set appropriate values for your environment.

TCP port of the XML data transfer: select an appropriate TCP port. The LIOS assigned default value is 9974 which should be suitable for most installations.

Enable TLS Protocol: check this option for an encrypted connection (see section 2.4 on page 6 for details).

Buffering of the CF card: select the appropriate data buffering mode (see section 3 on page 7 for details).

Transmitted data: select the data which shall be included in the XML files transmitted by the XML Data and Control plug-in.

Note: *Backscattering data* includes the backscatter profile, the attenuation data along the fibre and the OTDR profile (see sections 4.2 on page 9 and 4.3 on page 15 for details). It is available only if the transmission is enabled in the measurement settings. Attenuation and OTDR data is not available for fire detection (LHD) systems. See the *LHD3-/OTS3-Series Controllers Operation Manual* for details.

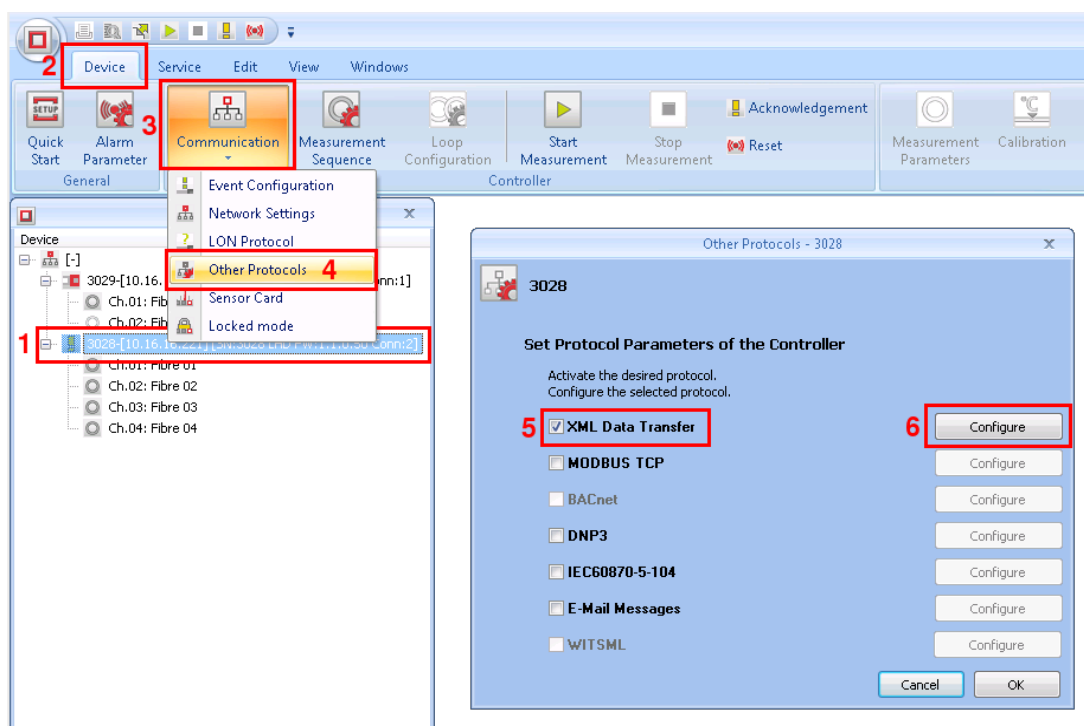


Figure 5: Activation of the XML Data and Control plug-in

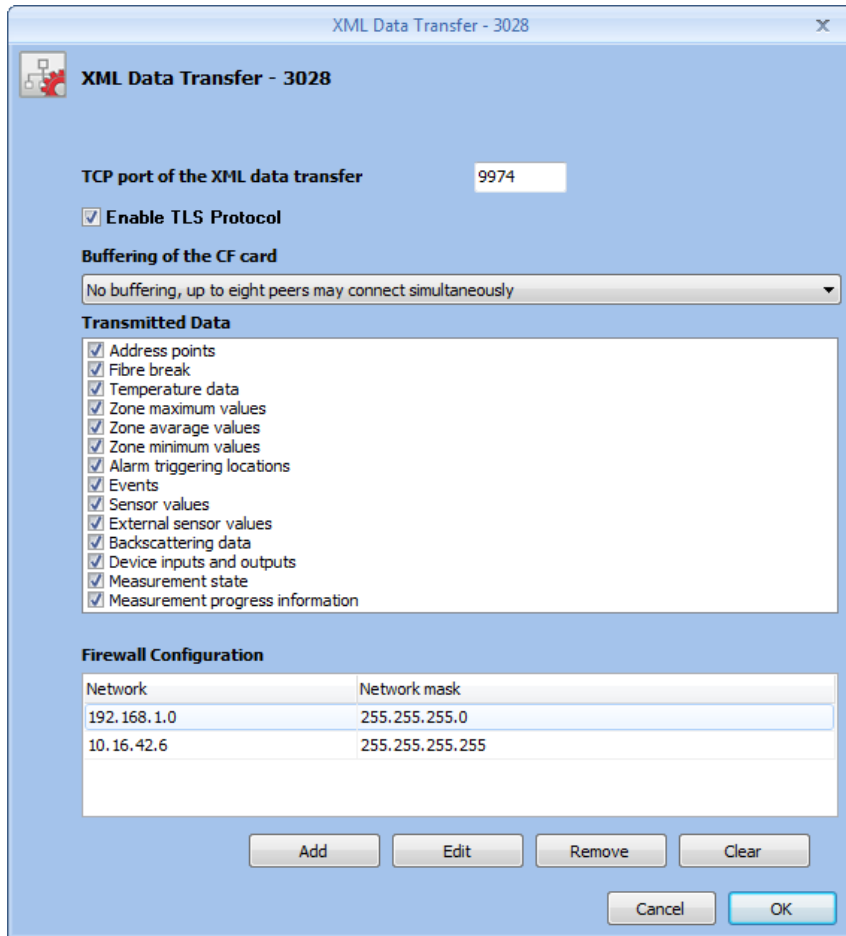


Figure 6: XML Data and Control plug-in configuration

Firewall configuration: define the firewall configuration for the XML Data and Control plug-in. If the list is empty, a peer with any IP address is permitted to connect the XML Data and Control plug-in. Otherwise, the peer's IP address must match at least one of the *Network* and *Network Mask* pairs from the list to access the XML Data and Control plug-in. Examples:

- 192.168.1.0/255.255.255.0: all peers with IP address 192.168.1.0 up to 192.168.1.255, inclusively, may connect the XML Data and Control plug-in (in other words, the network mask checks that the first three values of the IP address match, while the fourth value is irrelevant);
- 10.16.42.6/255.255.255.255: the peer station with IP address 10.16.42.6 may connect the XML Data and Control plug-in (in other words, the network mask checks that the IP address is exactly one value).

Up to three network/network mask pairs can be configured. Ask your network administrator about the proper configuration of the firewall.

Click the *Add* button to add a new network/network mask pair to the list. Select an entry from the list, and click *Edit* or *Remove* to edit the selected entry or to remove it from the list, respectively. Click *Clear* to remove all entries from the list.

Note: This firewall configuration is completely independent from the OTS Controller's firewall configuration and *only* influences the access to the XML Data and Control plug-in running on the OTS controller.



A LIOS XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- $Id: LIOSXml.xsd 586 2013-09-23 10:09:14Z adress $ -->
<xsd:schema targetNamespace="http://www.lios-support.com/LIOSXml"
  elementFormDefault="unqualified" attributeFormDefault="unqualified"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:lios="http://www.lios-support.com/LIOSXml">

  <xsd:complexType name="LIOSxmlT">
    <xsd:choice minOccurs="1" maxOccurs="1">
      <xsd:element name="measData" type="lios:LIOSMeasDataT" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="systemData" type="lios:LIOSSysDataT" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="statusData" type="lios:LIOSStatDataT" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="command" type="lios:LIOSCommandT" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="progress" type="lios:LIOSProgressT" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="result" type="lios:resultT" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="config" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="schema" type="xsd:base64Binary" minOccurs="1" maxOccurs="1"/>
    </xsd:choice>
    <xsd:attribute name="version" type="xsd:string" use="required" fixed="1.8"/>
  </xsd:complexType>

  <xsd:element name="LIOSxml" type="lios:LIOSxmlT"/>

  <xsd:complexType name="LIOSMeasDataT">
    <xsd:sequence minOccurs="1" maxOccurs="1">
      <xsd:element name="timeStamp" type="lios:dateTimeNtpT" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Controller time stamp when the measurement data record has
            been created. The element is missing if no measurement data is
            available.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="breakPos" type="xsd:float" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Position of a fibre break in m relative to the reference point.
            If the element is not included, this indicates that no fibre break has been
            detected. Note that this element is used only for a fibre-break in single-ended
            mode, and never used for fibres in loop mode.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="breakRange" type="lios:breakRangeT" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Fibre break range in loop mode. If the element is not included,
            this indicates that no fibre break has been detected. Note that this element is
            never transmitted for fibres in single-ended mode.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="addrPts" type="lios:addressPtDataT" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>List of triggered address points and flags.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="profile" type="lios:profileT" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Temperature profile.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="bscat" type="lios:bscatT" maxOccurs="1" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Backscatter profile.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="da" type="lios:posValT" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Delta-Alpha profile. The values are
            non-dimensional.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```



```
<xsd:element name="otdr" type="lios:posValT" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>OTDR profile. The values are
      non-dimensional.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="roc" type="lios:posValT" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Rate-of-change profile. The values are given in Kelvin per
      second.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="zones" type="lios:zonesT" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>List of zone values.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="alarmLocs" type="lios:alarmLocListT" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>List of alarm triggering locations.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="otsSerial" type="xsd:string" use="required">
  <xsd:annotation>
    <xsd:documentation>OTS serial number, assigned by LIOS
      Technology.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="otsModel" type="xsd:string" use="required">
  <xsd:annotation>
    <xsd:documentation>OTS model.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="otsName" type="xsd:string" use="required">
  <xsd:annotation>
    <xsd:documentation>User-defined name of the OTS controller.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="fibre" type="xsd:nonNegativeInteger" use="required">
  <xsd:annotation>
    <xsd:documentation>Logical fibre number, starting with 0, to which the measurement
      data packet belongs.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="fibreName" type="xsd:string" use="required">
  <xsd:annotation>
    <xsd:documentation>User-defined name of the fibre.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>

<xsd:complexType name="LIOSSysDataT">
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="timeStamp" type="lios:dateTimeNtpT" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Controller time stamp when the system data record has been
          created.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="addrPts" type="lios:addressPtDataT" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="event" type="xsd:int" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Event code.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="inputState" type="xsd:hexBinary" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Bit-packed states of the programmable
          inputs.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```



```
</xsd:element>
<xsd:element name="outputState" type="lios:outputStateT" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Bit-packed states of the programmable
      outputs.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="intSens" type="lios:intSensT" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Sensor readings from the built-in sensors of the
      device.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="extSens" type="lios:extSensT" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Sensor readings from the optional sensors add-on
      card.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="otsSerial" type="xsd:string" use="required">
  <xsd:annotation>
    <xsd:documentation>OTS serial number, assigned by LIOS
      Technology.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="otsModel" type="xsd:string" use="required">
  <xsd:annotation>
    <xsd:documentation>OTS model.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="otsName" type="xsd:string" use="required">
  <xsd:annotation>
    <xsd:documentation>User-defined name of the OTS controller.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>

<xsd:complexType name="LIOSStatDataT">
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="timeStamp" type="lios:dateTimeNtpT" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Controller time stamp when the status data has been
          created.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="inputState" type="xsd:hexBinary" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Bit-packed states of the programmable
          inputs.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="outputState" type="lios:outputStateT" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Bit-packed states of the programmable
          outputs.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="measState" type="lios:measStateT" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Measurement and alarm processing status.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="statistics" type="lios:statisticsT" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Statistical data of the controller, collected since the last
          statistics reset.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="fibre" type="lios:fibreStatT" minOccurs="1" maxOccurs="16">
      <xsd:annotation>
```




```
        <xsd:documentation>Status of all fibres.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="cfCard" type="lios:cfStatT" maxOccurs="1" minOccurs="1">
    <xsd:annotation>
        <xsd:documentation>Status of the CF memory card.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="otsSerial" type="xsd:string" use="required">
    <xsd:annotation>
        <xsd:documentation>OTS serial number, assigned by LIOS
            Technology.</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="otsModel" type="xsd:string" use="required">
    <xsd:annotation>
        <xsd:documentation>OTS model.</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="otsName" type="xsd:string" use="required">
    <xsd:annotation>
        <xsd:documentation>User-defined name of the OTS controller.</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
</xsd:complexType>

<xsd:complexType name="LIOSCommandT">
    <xsd:simpleContent>
        <xsd:extension base="lios:commandValsT">
            <xsd:attribute name="arg" type="xsd:string" use="optional">
                <xsd:annotation>
                    <xsd:documentation>Optional argument of the command.</xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="LIOSProgressT">
    <xsd:annotation>
        <xsd:documentation>Fibre measurement progress information, given in
            percent.</xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
        <xsd:extension base="xsd:float">
            <xsd:attribute name="otsSerial" type="xsd:string" use="required">
                <xsd:annotation>
                    <xsd:documentation>OTS serial number, assigned by LIOS
                        Technology.</xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="otsModel" type="xsd:string" use="required">
                <xsd:annotation>
                    <xsd:documentation>OTS model.</xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="otsName" type="xsd:string" use="required">
                <xsd:annotation>
                    <xsd:documentation>User-defined name of the OTS controller.</xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="fibre" type="xsd:nonNegativeInteger" use="required">
                <xsd:annotation>
                    <xsd:documentation>Logical fibre number, starting with 0, to which the
                        progress data packet belongs.</xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="fibreName" type="xsd:string" use="required">
                <xsd:annotation>
                    <xsd:documentation>User-defined name of the fibre.</xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```



```

    </xsd:annotation>
  </xsd:attribute>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="resultT">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="ok" type="xsd:boolean" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="zoneValT">
  <xsd:annotation>
    <xsd:documentation>Maximum, average and minimum temperature data in a
      zone.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="tMin" type="lios:tAtPosT" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Minimum temperature in degrees C within the zone and position
          relative to the reference point where the temperature has been detected. For a
          zone which is completely located behind a fibre break, the position and value
          are both NaN.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="tAvg" type="xsd:float" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Average temperature in degrees C within the zone. For a zone
          which is completely located behind a fibre break, the value is
          NaN.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="tMax" type="lios:tAtPosT" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Maximum temperature in degrees C within the zone and position
          relative to the reference point where the temperature has been detected. For a
          zone which is completely located behind a fibre break, the position and value
          are both NaN.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="idx" type="xsd:nonNegativeInteger" use="required">
    <xsd:annotation>
      <xsd:documentation>Zone index, counting from 0.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="tAtPosT">
  <xsd:annotation>
    <xsd:documentation>Temperature in degrees C at a position, measured in
      m.</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:float">
      <xsd:attribute name="pos" type="xsd:float">
        <xsd:annotation>
          <xsd:documentation>Position in m relative to the reference point at which the
            temperature value has been measured.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="floatTimeT">
  <xsd:annotation>
    <xsd:documentation>Float value, measured at a certain time.</xsd:documentation>
  </xsd:annotation>
</xsd:complexType>

```



```
</xsd:annotation>
<xsd:simpleContent>
  <xsd:extension base="xsd:float">
    <xsd:attribute name="time" type="xsd:dateTime">
      <xsd:annotation>
        <xsd:documentation>Time stamp when the float value has been
          acquired.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="dateTimeNtpT">
  <xsd:annotation>
    <xsd:documentation>Time stamp with NTP status.</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:dateTime">
      <xsd:attribute name="ntp" type="xsd:boolean" use="optional">
        <xsd:annotation>
          <xsd:documentation>Indicate that the time stamp is accurate, i.e. the OTS
            controller is synchronised with the configured NTP server. A missing attribute
            implies 'false'.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="alarmLocT">
  <xsd:annotation>
    <xsd:documentation>Range of alarm triggering locations in m. If only the start element
      is present, this indicates an isolated (single-point) trigger. Otherwise, the values
      indicate the range.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="start" type="xsd:float" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="end" type="xsd:float" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="profileT">
  <xsd:annotation>
    <xsd:documentation>Temperature profile. Note that any part of the temperature profile
      which is located behind a fibre break will be omitted. Thus, the number of profile
      elements will be smaller than thecount in case of a fibre break.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="tVal" type="lios:tAtPosT" minOccurs="1" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Position in m relative to the reference point and the measured
          temperature value in degrees C.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="count" type="xsd:nonNegativeInteger" use="required">
    <xsd:annotation>
      <xsd:documentation>Number of data points in the temperature
        profile.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="tRef" type="xsd:float" use="required">
    <xsd:annotation>
      <xsd:documentation>Reference temperature in degrees C.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="addressPtDataT">
  <xsd:annotation>
```



```
<xsd:documentation>List of triggered fibre address points.</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
  <xsd:element name="ap" type="lios:addressPtT" minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="state" type="lios:addressPtMaskT" minOccurs="1" maxOccurs="1">
    <xsd:annotation>
      <xsd:documentation>The value is a bit mask, indicating the current states of all
        address points.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="zonesT">
  <xsd:annotation>
    <xsd:documentation>List of zones.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="zone" type="lios:zoneValT" minOccurs="1" maxOccurs="1000"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="alarmLocListT">
  <xsd:sequence>
    <xsd:element name="loc" type="lios:alarmLocT" minOccurs="1" maxOccurs="106"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="intSensT">
  <xsd:sequence>
    <xsd:element name="devTemp" type="xsd:float" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Internal device temperature in degrees C.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="devHumid" type="xsd:float" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Internal humidity of the device in % rel. humidity. Note that
          the value is NaN if the device is not equipped with an internal humidity
          sensor.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="inVolt" type="xsd:float" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Device power supply input voltage in V DC.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="extSensT">
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="sens" type="lios:floatIdxT" minOccurs="6" maxOccurs="6"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="floatIdxT">
  <xsd:annotation>
    <xsd:documentation>Float value at an index.</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:float">
      <xsd:attribute name="idx" type="xsd:nonNegativeInteger" use="required">
        <xsd:annotation>
          <xsd:documentation>Value index.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```



```
<xsd:complexType name="addressPtT">
  <xsd:annotation>
    <xsd:documentation></xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:unsignedShort">
      <xsd:attribute name="flags" type="xsd:unsignedShort" use="required">
        <xsd:annotation>
          <xsd:documentation>Address point flags.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="commandValsT">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="reset"/>
    <xsd:enumeration value="ack"/>
    <xsd:enumeration value="start"/>
    <xsd:enumeration value="stop"/>
    <xsd:enumeration value="time"/>
    <xsd:enumeration value="status"/>
    <xsd:enumeration value="schema"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="addressPtMaskT">
  <xsd:restriction base="xsd:hexBinary">
    <xsd:maxLength value="776"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="bscatT">
  <xsd:annotation>
    <xsd:documentation>Backscatter profile</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="bsVal" type="lios:bsAtPosT" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="count" type="xsd:nonNegativeInteger"/>
</xsd:complexType>

<xsd:complexType name="bsAtPosT">
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="s" type="xsd:float">
      <xsd:annotation>
        <xsd:documentation>Stokes values</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="as" type="xsd:float">
      <xsd:annotation>
        <xsd:documentation>Antistokes value</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="pos" type="xsd:float">
    <xsd:annotation>
      <xsd:documentation>Position in m relative to the device's fibre connector at which
        the backscatter value has been measured.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="outputStateT">
  <xsd:annotation>
    <xsd:documentation>logical states of the fixed and programmable device
      outputs</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:hexBinary">
      <xsd:attribute name="sysFault" type="xsd:boolean" use="required">
```



```
<xsd:annotation>
  <xsd:documentation>indicates if the system fault output is logically
    active</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="sysAlarm" type="xsd:boolean" use="required">
  <xsd:annotation>
    <xsd:documentation>indicates if the system alarm output is logically
      active</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="breakRangeT">
  <xsd:sequence>
    <xsd:element name="start" type="xsd:float">
      <xsd:annotation>
        <xsd:documentation>Start of the "dead area" caused by a fibre break in loop
          mode.</xsd:documentation>
      </xsd:annotation></xsd:element>
    <xsd:element name="end" type="xsd:float">
      <xsd:annotation>
        <xsd:documentation>End of the "dead area" caused by a fibre break in loop
          mode.</xsd:documentation>
      </xsd:annotation></xsd:element>
    </xsd:sequence>
  </xsd:complexType>

<xsd:complexType name="statisticsT">
  <xsd:annotation>
    <xsd:documentation>Statistical data of the OTS controller. Note that fields are
      omitted after a reset of the user statistics when new new data has been collected
      (about 30 seconds).</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="runtime" type="xsd:nonNegativeInteger" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Device run time in seconds.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="tMin" type="lios:floatTimeT" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Minimum internal device temperature in degrees
          C.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="tMax" type="lios:floatTimeT" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Maximum internal device temperature in degrees
          C.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="humid" type="lios:floatTimeT" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Maximum internal device humidity in %.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="voltMin" type="lios:floatTimeT" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Minimum device DC input voltage in V.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="voltMax" type="lios:floatTimeT" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Maximum device DC input voltage in V.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```



```
<xsd:complexType name="fibreStatT">
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="commissioned" type="xsd:boolean" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Indicates if the fibre has been
          commissioned.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="length" type="xsd:float" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Detected fibre length, present only if the fibre has been
          commissioned.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="index" type="xsd:nonNegativeInteger" use="required">
    <xsd:annotation>
      <xsd:documentation>Logical fibre number, starting with 0.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="fibreName" type="xsd:string" use="required">
    <xsd:annotation>
      <xsd:documentation>User-defined name of the fibre.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="cfStatT">
  <xsd:annotation>
    <xsd:documentation>Status information of the CF card.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="size" type="xsd:nonNegativeInteger" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Size of the CF card in kBytes.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="free" type="xsd:nonNegativeInteger" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Free space in kBytes.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="smart" type="xsd:boolean" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>True if the S.M.A.R.T. logic of the device reports a good
          health state. The element is omitted if the device does not support
          S.M.A.R.T.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="measStateT">
  <xsd:annotation>
    <xsd:documentation>Measurement and alarm processing status
      information.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="activeFibre" type="xsd:int" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>If a measurement is active, the currently measured fibre,
          starting with 0 for the first channel. If no regular measurement (i.e. a
          sequence measurement or a single-fibre measurement) is active, the value is
          -1.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="testMode" type="xsd:boolean" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Indicates if the system is in test mode.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```



```
</xsd:element>
<xsd:element name="alarmProcessing" type="xsd:boolean" minOccurs="1" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Indicates if the alarm processing is configured properly and if
      all fibres in the measurement sequence have a zone configuration which means
      that they can produce alarm, pre-alarm or fibre break address
      points.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="posValT">
  <xsd:annotation>
    <xsd:documentation>Profile of values.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="posVal" type="lios:valAtPosT" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="count" type="xsd:nonNegativeInteger"></xsd:attribute>
</xsd:complexType>

<xsd:complexType name="valAtPosT">
  <xsd:annotation>
    <xsd:documentation>Value at a position, measured in m. The dimension of the value
      is defined by the parent container.</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:float">
      <xsd:attribute name="pos" type="xsd:float">
        <xsd:annotation>
          <xsd:documentation>Position in m relative to the reference point at which the
            value has been measured or calculated.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

</xsd:schema>
```

B Examples

B.1 Fibre measurement data

```
<?xml version="1.0" encoding="UTF-8"?>
<lios:LIOSXml xmlns:lios="http://www.lios-support.com/LIOSXml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.lios-support.com/LIOSXml LIOSXml.xsd"
  version="1.8">
  <measData fibre="1" otsModel="OTS3010" otsSerial="3028" otsName="Tunnel north"
    fibreName="Tunnel Tube 2">
    <timeStamp ntp="1">2012-08-21T11:52:17.627Z</timeStamp>
    <addrPts>
      <ap flags="769">1003</ap>
      <ap flags="769">1004</ap>
      <ap flags="769">1005</ap>
      <ap flags="769">1006</ap>
      <ap flags="769">1007</ap>
      <ap flags="257">8</ap>
      <ap flags="769">1008</ap>
      <ap flags="257">9</ap>
      <ap flags="769">1009</ap>
      <ap flags="1281">3003</ap>
      <ap flags="1537">3009</ap>
      <state>0000000000000000000000000208000[...0000000000300</state>
      <!-- Note: values in [...] are stripped -->
    </addrPts>
```




```
<profile count="509" tRef="31.0063">
  <tVal pos="0">33.669</tVal>
  <tVal pos="1">34.1595</tVal>
  <tVal pos="2">34.0124</tVal>
  <tVal pos="3">33.8213</tVal>
  <tVal pos="4">33.9677</tVal>
  <tVal pos="5">34.0269</tVal>
  <tVal pos="6">33.8453</tVal>
  <tVal pos="7">33.6647</tVal>
  <tVal pos="8">33.5849</tVal>
  <tVal pos="9">33.8755</tVal>
  <tVal pos="10">34.0562</tVal>
  <!-- Note: more temperature profile values stripped -->
  <tVal pos="506">65.0513</tVal>
  <tVal pos="507">65.5898</tVal>
  <tVal pos="508">65.3389</tVal>
</profile>
<bscat count="2290">
  <bsVal pos="-125">
    <s>7093.2</s>
    <as>5849.59</as>
  </bsVal>
  <bsVal pos="-124">
    <s>6738.9</s>
    <as>4671.11</as>
  </bsVal>
  <bsVal pos="-123">
    <s>6549.94</s>
    <as>4805.82</as>
  </bsVal>
  <bsVal pos="-122">
    <s>6576.57</s>
    <as>4813.93</as>
  </bsVal>
  <bsVal pos="-121">
    <s>6581.67</s>
    <as>4821.2</as>
  </bsVal>
  <bsVal pos="-120">
    <s>6569.89</s>
    <as>4787.85</as>
  </bsVal>
  <bsVal pos="-119">
    <s>6552.27</s>
    <as>4808.66</as>
  </bsVal>
  <bsVal pos="-118">
    <s>6554.22</s>
    <as>4793.37</as>
  </bsVal>
  <bsVal pos="-117">
    <s>6541.08</s>
    <as>4779.7</as>
  </bsVal>
  <!-- Note: more backscatter profile values stripped -->
  <bsVal pos="1018">
    <s>8.24343</s>
    <as>-5.96501</as>
  </bsVal>
  <bsVal pos="1019">
    <s>8.6862</s>
    <as>-15.8135</as>
  </bsVal>
</bscat>
<zones>
  <zone idx="0">
    <tMin pos="12">33.5574</tMin>
    <tAvg>34.6261</tAvg>
    <tMax pos="50">36.1034</tMax>
  </zone>
  <zone idx="1">
```



```
<tMin pos="57">36.0392</tMin>
<tAvg>37.7511</tAvg>
<tMax pos="94">39.3762</tMax>
</zone>
<zone idx="2">
  <tMin pos="101">39.1313</tMin>
  <tAvg>40.7211</tAvg>
  <tMax pos="152">42.7019</tMax>
</zone>
<!-- Note: more zone values stripped -->
<zone idx="9">
  <tMin pos="455">61.3627</tMin>
  <tAvg>63.0354</tAvg>
  <tMax pos="501">64.747</tMax>
</zone>
</zones>
<alarmLocs>
  <loc>
    <start>416</start>
    <end>505</end>
  </loc>
</alarmLocs>
</measData>
</lios:LIOSEXml>
```

B.2 Device status information data

```
<?xml version="1.0" encoding="UTF-8"?>
<lios:LIOSEXml xmlns:lios="http://www.lios-support.com/LIOSEXml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.lios-support.com/LIOSEXml LIOSEXml.xsd"
  version="1.8">
  <statusData otsModel="OTS3010" otsSerial="3028" otsName="Test Fire Detection">
    <timeStamp ntp="1">2012-11-08T14:25:37.313Z</timeStamp>
    <inputState>0000000000</inputState>
    <outputState sysFault="0" sysAlarm="0">0000000000000000000000000000</outputState>
    <measState>
      <activeFibre>0</activeFibre>
      <testMode>1</testMode>
      <alarmProcessing>1</alarmProcessing>
    </measState>
    <statistics>
      <runtime>1576126</runtime>
      <tMin time="2012-10-01T06:36:41">19</tMin>
      <tMax time="2012-09-21T14:15:37">37</tMax>
      <humid time="2012-09-20T06:22:00">35.7</humid>
      <voltMin time="2012-09-20T06:20:00">23.8844</voltMin>
      <voltMax time="2012-09-26T08:31:50">24.0578</voltMax>
    </statistics>
    <fibre index="0" fibreName="North">
      <commissioned>1</commissioned>
      <length>3955</length>
    </fibre>
    <fibre index="1" fibreName="South">
      <commissioned>1</commissioned>
      <length>3953</length>
    </fibre>
    <fibre index="2" fibreName="3">
      <commissioned>1</commissioned>
      <length>0</length>
    </fibre>
    <fibre index="3" fibreName="4">
      <commissioned>1</commissioned>
      <length>0</length>
    </fibre>
    <cfCard>
      <size>1995952</size>
      <free>399120</free>
      <smart>1</smart>
    </cfCard>
```



```
</statusData>  
</lios:LIOSSml>
```



page intentionally left blank



page intentionally left blank

This Page Left Intentionally Blank

Linear Optical Sensors



COMMUNICATION MANUAL PART VI: E-MAIL NOTIFICATIONS

OTS3- / LHD3-Series Controllers



OTS3- / LHD3-Series Controllers
Communication Manual Part VI: E-Mail Notifications
Revision 1.1

Revision History:

Rev.	Date	Changes
1.0	07-11-2012	initial release
1.1	23-06-2014	update standards references

Property of LIOS Technology GmbH, Köln.

This document and its associated distribution media contain proprietary information of LIOS Technology GmbH and may not be used, copied, or distributed in any form without the written permission of LIOS Technology GmbH.

Copies of this document may be made for internal use by the legally licensed owner.

Copyright © LIOS Technology GmbH 2012 – 2014. All rights reserved.

LIOS Technology GmbH
Schanzenstraße 39
Building D9–D13
D-51 063 Köln
Germany
Phone (+49) 221 99887 0
www.lios-tech.com
E-Mail: info@lios-tech.com



Contents

1 Introduction	5
2 Applicable standards	5
2.1 Mail Transfer Agent	5
2.2 E-Mail messages	5
2.3 Security considerations	5
3 Transmitted data	5
3.1 Subject	5
3.2 Header	6
3.3 Alarms	6
3.4 Faults	7
3.5 Pre-alarms	7
3.6 Warnings	8
4 Configuration	8

List of Figures

1 Activation of the E-Mail Notifications plug-in	8
2 E-Mail Notifications configuration	9



page intentionally left blank



1 Introduction

The E-Mail Notifications option for the OTS3-/LHD3-Series Controllers sends standard e-mails when an error, warning, alarm or pre-alarm occurs. The triggering sources can be configured by the user.

Warning: E-Mail notifications may be used for informative purposes **ONLY**. As an arbitrary delay can be introduced by Mail Transfer Agents (MTA's) and other network components between sending the messages and their reception, they **MUST NOT** be used as primary notification method for critical events.

2 Applicable standards

2.1 Mail Transfer Agent

Any Mail Transfer Agent (MTA) which complies with the *Simple Mail Transfer Protocol* (SMTP, [7]) can be used with the e-mail notification option.

Optionally, the "PLAIN" or "LOGIN" authentication mechanisms [6] are supported if the MTA requires authentication. In this case, the e-mail notification plug-in will automatically detect whether the MTA supports *Secure SMTP over Transport Layer Security* ("STARTTLS", [4]), and if possible perform the transaction over an encrypted channel as to protect the authentication data.

Note that E-Mail Notifications plug-in will *not* buffer any e-mail messages in the OTS controller. If the MTA rejects the message by issuing a negative reply, the message will be lost, and the reason returned by the MTA is stored in the OTS controller's event memory. This does also apply to transient negative completion replies of the MTA (code 4xx) which are used as "greylisting" technique in some installations.

2.2 E-Mail messages

The E-Mail Notifications plug-in sends standard single-part text messages according to the *Internet Message Format* standard [2, 8]. All parts may include Unicode (utf-8) encoded non-ASCII characters [1, 3]. Long lines are formatted according to [5] for an improved presentation.

2.3 Security considerations

The E-Mail Notifications plug-in does not support using SSL encrypted SMTP connections to port 465. However, whenever authentication data shall be transmitted to the MTA, the e-mail notification plug-in will try to switch to a "STARTTLS" encrypted connection according to [4] if the MTA supports it.

If no authentication is configured, the connection will always be unencrypted, even if the MTA advertises "STARTTLS".

3 Transmitted data

3.1 Subject

The message subject always contains the following information:

- OTS model and serial number
- the user-defined name of the device
- the total numbers of alarms, faults, pre-alarms and warnings. Categories with no events are omitted.



3.2 Header

The message body contains a header including the following information:

- Project information:
 - the user-defined project name
 - the user-defined project description
- Device information:
 - OTS model and serial number
 - the user-defined name of the device
 - the user-defined description of the device
- the date and time in UTC (Universal Time Coordinated) when the message has been generated
- a note if the E-Mail Notifications plug-in has been configured to suppress sending further messages until the *Acknowledgement* operation has been performed.

Example:

Project

Name: M14 Fire Detection
Description: Fire Detection Tunnel M14 (north and south)

Device

Model: OTS3010 S/N 3028
Name: Tunnel South Park
Description: Fire Detection Section R2

Time (UTC): Wed 17 Oct 2012 15:06:04

Note: For further details about the status of the controller and the issues below, please check its event memory.

Note: No further e-mail will be transmitted before the Acknowledgement operation of the controller has been performed.

3.3 Alarms

The ALARMS section includes all system and fibre alarm address points. For fibre alarm address points, the channel number as well as the user-defined name and description are included. Each address point information includes the user-defined names of the zone.

Likewise, for system alarm inputs, the user-defined input names are included. Example:

ALARMS
=====

System alarm address points: 1
Fibre alarm address points: 1

System Address Points



```
Address point #1: Alarm Input #1: Extinguisher 9

Fibre Address Points
-----

Channel #:    1
Name:        North
Description: Tunnel tube #1 (north lane)

Address point #8: Alarm in zone #8: Section V13
```

3.4 Faults

The FAULTS section includes all system and fibre fault events as well as the address points related to them. For fibre-related faults, the fibre information is included as for alarms (see section 3.3 on the facing page). Example:

```
FAULTS
=====

System fault events and address points: 3
Fibre fault events and address points:  0

Events
-----

System: Measurement stopped
Measurement stopped: The measurement was stopped by a command

System Address Points
-----

Address point #101: System fault

Address point #109: Measurement stopped
```

3.5 Pre-alarms

The PRE-ALARMS section is similar to the fibre-related part of the ALARM section, but contains pre-alarm information. Example:

```
PRE-ALARMS
=====

Fibre pre-alarm address points:  1

Fibre Address Points
-----

Channel #:    2
Name:        South
Description: Tunnel tube #2 (south lane)

Address point #1005: Pre-Alarm in zone #5: Section X4
```



3.6 Warnings

The WARNINGS section includes all system and fibre warning events similar to the FAULTS section (see section 3.4 on the previous page). Example:

```
WARNINGS
=====
```

```
System warning events and address points: 2
Fibre warning events:                      0
```

```
Events
-----
```

```
System: Measurement warning
CF card full: The CF card is too full, data is lost or erased
```

```
System Address Points
-----
```

```
Address point #202: Measurement warning
```

4 Configuration

In order to use the E-Mail Notifications plug-in, it first has to be activated and configured (see figure 1)

1. Select the OTS controller in the tree view
2. Activate the *Device* tab.

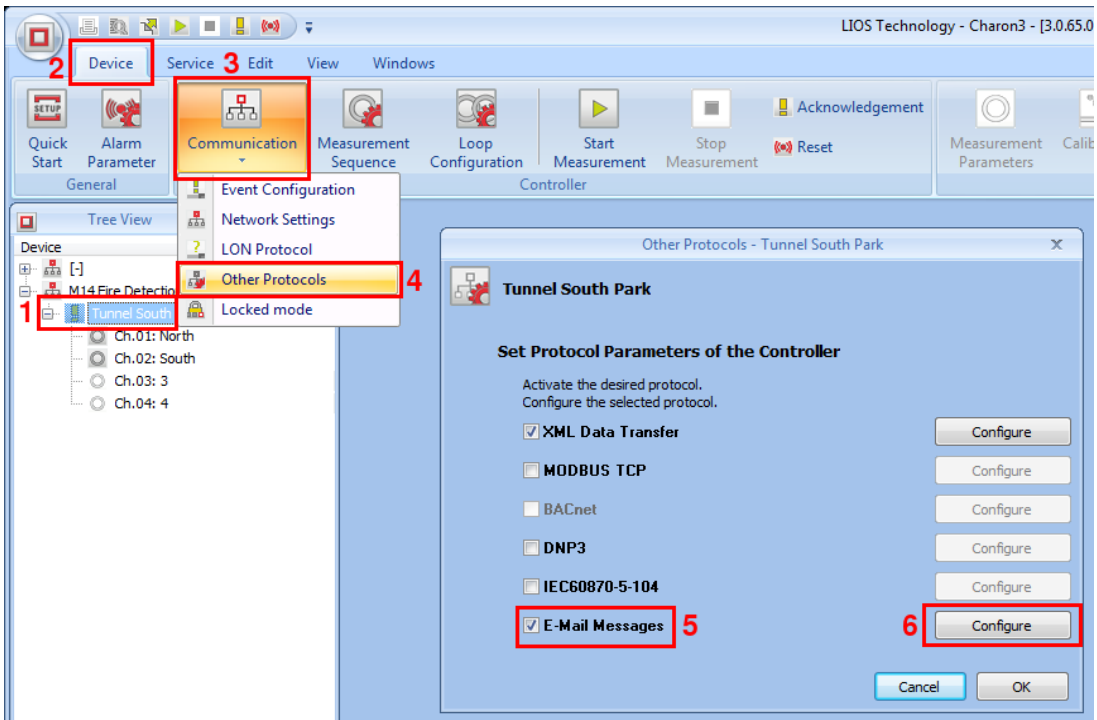


Figure 1: Activation of the E-Mail Notifications plug-in



3. In the *Controller* group, click *Communication*.
4. Select *Other Protocols* from the menu.
5. in the *Other Protocols* dialogue, activate the e-mail notification plug-in using the check box.
6. Click *Configure* to configure the e-mail notification options.

In the e-mail notification options dialogue (see figure 2), set appropriate values for your environment.

SMTP Server: enter the IP address or the Full Qualified Domain Name (FQDN) of your organisation's SMTP server. The latter may be used only if the Domain Name System (DNS) server is configured properly. See the *LHD3-/OTS3-Series Controllers Operation Manual* for details about a proper DNS configuration.

SMTP Server Port: enter the TCP port number where the SMTP server listens. Typical values are 25 ("SMTP") or 587 ("Submission").

SMTP user name: if your organisation's SMTP server requires authentication, enter the user name in this field. Simply leave this field empty if no authentication is required.

Password: if your organisation's SMTP server requires authentication, enter the password for the user name entered in the previous field.

Mail Sender: enter the e-mail address which shall be used as sender in any message generated by the OTS controller. Note that the e-mail address must be a RFC 5321/5322 compliant mailbox like `some.sender@company.com` [7, 8].

Recipient: enter up to three e-mail addresses to which any message generated by the OTS controller shall be transmitted. Note that all addresses must be a RFC 5321/5322 compliant mailboxes like `some.recipient@company.com` [7, 8].

Wait for an Acknowledgement: if this option is checked, the OTS controller will suppress the transmission of further messages after sending the first e-mail until an *Acknowledgement* operation has been performed. This protects the recipient's mailboxes from overflow if a larger number of events is

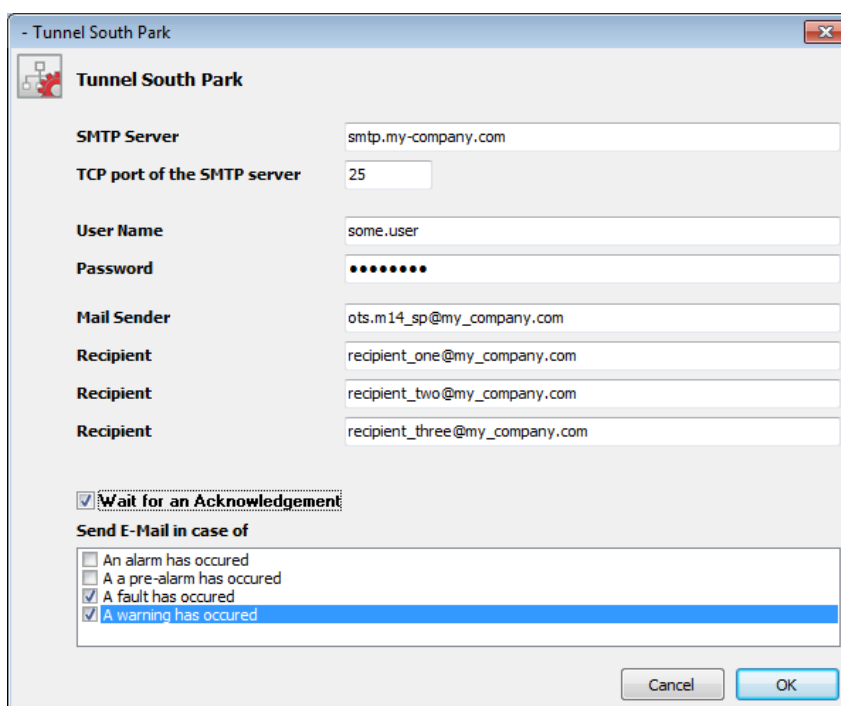


Figure 2: E-Mail Notifications configuration



expected. See the *LHD3-/OTS3-Series Controllers Operation Manual* for details about triggering an *Acknowledgement* operation.

Send E-Mail in case of: activate the conditions which shall trigger sending an e-mail. Note that at least one option must be selected.

References

- [1] N. FRIED and N. BORENSTEIN: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, November 1996 (IETF RFC 2045, <http://tools.ietf.org/html/rfc2045>).
- [2] N. FRIED and N. BORENSTEIN: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, November 1996 (IETF RFC 2046, <http://tools.ietf.org/html/rfc2046>).
- [3] K. MOORE: MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text, November 1996 (IETF RFC 2047, <http://tools.ietf.org/html/rfc2047>).
- [4] P. HOFFMAN: SMTP Service Extension for Secure SMTP over Transport Layer Security, February 2002 (IETF RFC 3207, <http://tools.ietf.org/html/rfc3207>).
- [5] R. GELLENS: The Text/Plain Format and DelSp Parameters, February 2004 (IETF RFC 3676, <http://tools.ietf.org/html/rfc3676>).
- [6] R. SIEMBORSKI and A. MELNIKOV: SMTP Service Extension for Authentication, July 2007. (IETF RFC 4954, <http://tools.ietf.org/html/rfc4954>).
- [7] J. KLENSIN: Simple Mail Transfer Protocol, October 2008 (IETF RFC 5321, <http://tools.ietf.org/html/rfc5321>).
- [8] P. RESNICK: Internet Message Format, October 2008 (IETF RFC 5322, <http://tools.ietf.org/html/rfc5322>).

— oOo —



page intentionally left blank



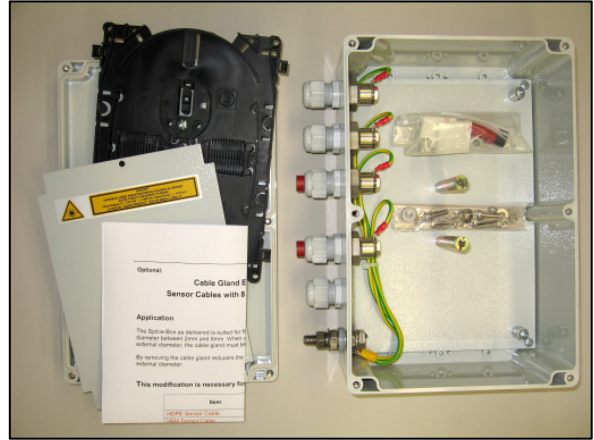
page intentionally left blank

LIOS SPLICE BOX WITH **GROUNDING SYSTEM**

Operations & Maintenance Manual
December 2015

Splice Box 4 Channels for Sensor Cable with Grounding System

Part number: 28024033



Specifications	
General description	Splice box with grounding system for stranded wire loose tube sensor cable of 4mm diameter "fast response". Installation in control cabinet close to OTS. Provides four entries for sensor cables, one entry for up to four pigtail cables and a grounding bolt. For use with OTS with multiple sensor cable channels.
Included items	<ul style="list-style-type: none"> - Splice cassette with cover and crimp protection sleeve holder, preinstalled in box - 3 red sealing plugs (1 spare), for unused sensor cable entries - 4 white sealing plugs (2 spare), for unused entries on quad sealing insert on pigtail cable entry - 1 grey cable gland reducer for pigtail cable entry - Set of screws, washers and nuts (M4) for mounting in cabinet
Connector for grounding	Bolt M8 with nut, wrench size 13mm
Sensor cable entries	Cable gland for cables of 2.5 – 5.5mm diameter, wrench size 20mm. Optional for cables of 5 – 9mm diameter when removing grey gland reducer. Cable gland equipped with special termination socket with spring contacts.
Pigtail cables entry (to OTS)	Cable gland, wrench size 20mm. Comes with quad sealing insert for up to four pigtail cables, each hole of 3mm diameter. Alternatively for metal free cables with cable gland for single cable of 2.5 – 5.5mm diameter or for single cable of 5 – 9mm diameter without grey gland reducer.
Outer diameter of sensor cable steel tube	Min. 1.8mm, max. 2mm
Transition resistance on spring contacts	Max. 0.5mΩ
Rated current of total grounding system	Min. 0.5kA for 1sec, 16A permanently
Material and colour box	Polycarbonate, halogen free, grey, RAL7035
Outer dimensions box (WxDxH)	240mm x 160mm x 90mm
Mounting box	4 holes in corners, 4.2mm diameter, distances: 228/130mm
Protection grade box	IP65
Weight	1.2kg
Storage / operating temperature	-40 °C – 80 °C
Compliances	Box: UL94-V2

LIOS Technology GmbH
 LINEAR OPTICAL SENSORS
 Schanzenstrasse 39
 Building D9-D13
 51063 Köln/Cologne Germany
 Tel.: +49 221 99887-0
 Fax: +49 221 99887-150
 info@lios-tech.com

Further information about our products and services can be obtained at our internet site www.lios-tech.com

MEMBER OF THE NKT GROUP

This document and all its components is protected by copyright. The recipient shall recognise these rights and shall not disclose the contents, either in whole or in part, of this document without the approval of the publisher to any other third party or use it for any other than its designated purpose.

This Page Left Intentionally Blank

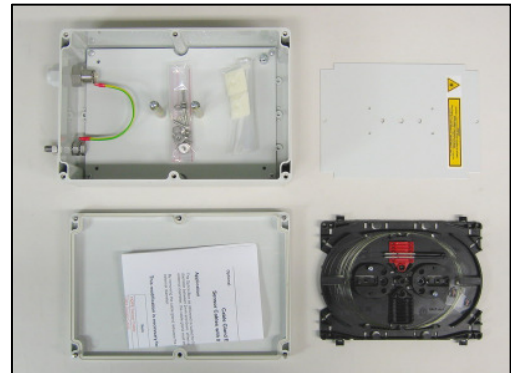
LIOS TERMINATION BOX **WITH GROUNDING SYSTEM**

Operations & Maintenance Manual
December 2015



Termination Box with Grounding System

Part number: 28024036



Specifications	
General description	Termination box with grounding system for sensor cable of 4mm diameter. Installation at the end of the sensor cable for low reflective termination of the end of the optical fibre. Arc fusion splicer required for connection of termination module.
Included items	<ul style="list-style-type: none"> - Splice cassette with cover and crimp protection sleeve holder, preinstalled in box - Termination module for fibre preinstalled in splice cassette - Set of screws, washers and nuts (M4) for mounting in cabinet
Type of termination module	Diffuse absorptive low reflective termination with angular cleaved fibre.
Fibre of termination module	GI50/125/250µm Suitable for multimode fibres GI50 or GI62.5
Length of fibre of term module in splice cass.	15m minimum
Peak of backscattered end reflection	< 0.1dB
Connector for grounding	Bolt M8 with nut, wrench size 13mm
Sensor cable entry	Cable gland for cables of 2 – 6mm diameter, wrench size 20mm Optional for cables of 4 – 8mm diameter when removing grey gland reducer. Cable gland equipped with special termination socket with spring contacts.
Outer diameter of sensor cable steel tube	Min. 1.8mm, max. 2mm
Transition resistance on spring contacts	Max. 0.5mΩ
Rated current of total grounding system	Min. 0.5kA for 1sec, 16A permanently
Material box	Polycarbonate, halogen free
Colour box	grey, RAL7035
Outer dimensions box (WxDxH)	240mm x 160mm x 90mm
Mounting box	4 holes in corners, 4.2mm diameter, distances: 228/130mm
Ingress protection grade box	IP65
Weight	1.2kg
Storage / operating temperature	-40 °C – 80 °C
Compliances	Box: UL94-V2

LIOS Technology GmbH
LINEAR OPTICAL SENSORS
Schanzenstrasse 39
Building D9-D13
51063 Köln/Cologne Germany
Tel.: +49 221 99887-0
Fax: +49 221 99887-150
info@lios-tech.com

Further information about our products and services can be obtained at our internet site
www.lios-tech.com

MEMBER OF THE NKT GROUP

This document and all its components is protected by copyright. The recipient shall recognise these rights and shall not disclose the contents, either in whole or in part, of this document without the approval of the publisher to any other third party or use it for any other than its designated purpose.

This Page Left Intentionally Blank

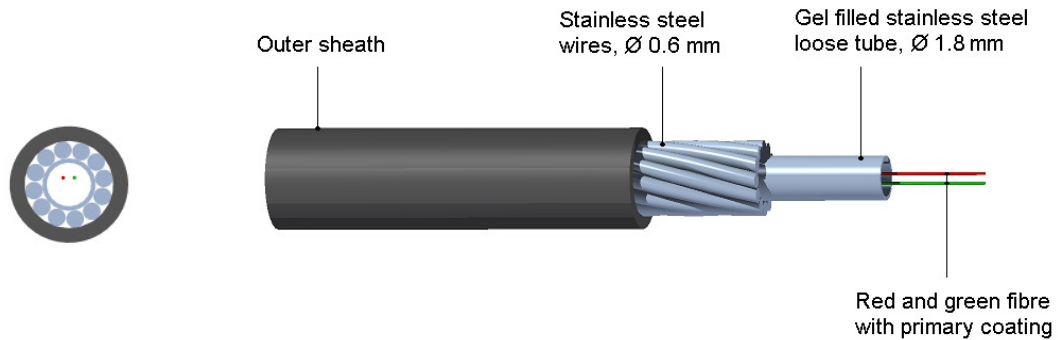
LIOS SENSOR CABLE, FAST **RESPONSE (TUNNEL)**

Operations & Maintenance Manual
December 2015

Sensor Cable Fast Response FRNC, 4.0mm

Part number: 28030504

Steel loose tube with 2 fibres, metal strength members and outer sheath:






Specifications

General description	The Sensor Cable complies with the manufacturer's guidelines of system calibration for the undisturbed operation as fire detection system.
----------------------------	--

Optical fibre	62,5/125µm, Graded index
----------------------	--------------------------

Mechanical Data

Steel loose tube	Hydrophobic gel filled as fibre protection, longitudinal water-tight.
Outer diameter	1.8 mm
Number of fibres	2
Armour	12 steel wires with 0.6 mm diameter
Outer sheath	FRNC, black
Outer diameter	4.0 mm
Labeling (white)	[Production lot]  [meter marking] LIOS Technology Fast Response FRNC 28030504  OPTICAL FIBRE CABLE 

Sensor Cable Fast Response FRNC, 4.0mm

Part number: 28030504

Mechanical characteristics

Min. bending radius	
without tensile loading (IEC 60794-1-2, G1)	40 mm
with tensile loading (IEC 60794-1-2, E6)	80 mm (500 cycles, 100 N)
Max. tensile strength (IEC 60794-1-2, E1)	
long term	1100 N
short term	2000 N
Weight	44 kg/km
Max. crush resistance (IEC 60794-1-2, E3)	10000 N/dm (10 min.)
Impact resistance (IEC 60794-1-2, E4)	20 impacts with 1.0 Nm (marking stamp 300 mm)

Thermal characteristics

Operating and storage temperature	- 25°C to + 85°C
Installation temperature	- 5°C to + 50°C
Short term temperature	- 50°C to + 150°C (max. 60 min.)

Flammability

Flame resistance	IEC 60332-1-2
Circuit Integrity with mechanical shock	EN 50200 – PH 90 (90 min, 830 °C, $\Delta\alpha < 3$ dB @ 1300 nm)
Smoke density	IEC 61034-2
Halogen free	IEC 60754-2
Fire load	0.16 MJ/m

Electrical characteristics

Sheath insulation (VDE 0276-620)	5000 Veff, DC
----------------------------------	---------------

Standards

Cable	EN 18700, IEC 60794
Fibre	IEC 60793

Remark	A calibration sheet can be supplied on request.
---------------	---

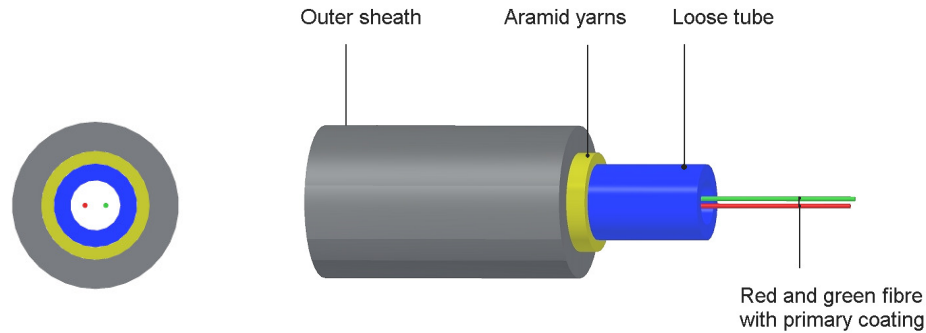
LIOS SENSOR CABLE, FAST RESPONSE (NON-TUNNEL)

**Operations & Maintenance Manual
December 2015**

Sensor Cable Metal Free FRNC, 4.0mm

Part number: 28030606

Loose tube with 2 fibres, aramid yarns and outer sheath:






Specifications

General description	The Sensor Cable complies with the manufacturer's guidelines of system calibration for the undisturbed operation as fire detection system.
----------------------------	--

Optical fibre	62,5/125µm, Graded index
----------------------	--------------------------

Mechanical Data

Loose tube	Hydrophobic gel filled as fibre protection, longitudinal water-tight.
Outer diameter	2.0 mm
Number of fibres	2
Armour	Aramid yarns as strain relief element
Outer sheath	FRNC, black
Outer diameter	4.0 mm
Labeling (white)	[Production lot]  [meter marking] LIOS Technology Metal Free FRNC 28030606  OPTICAL FIBRE CABLE 

Sensor Cable Metal Free FRNC, 4.0mm

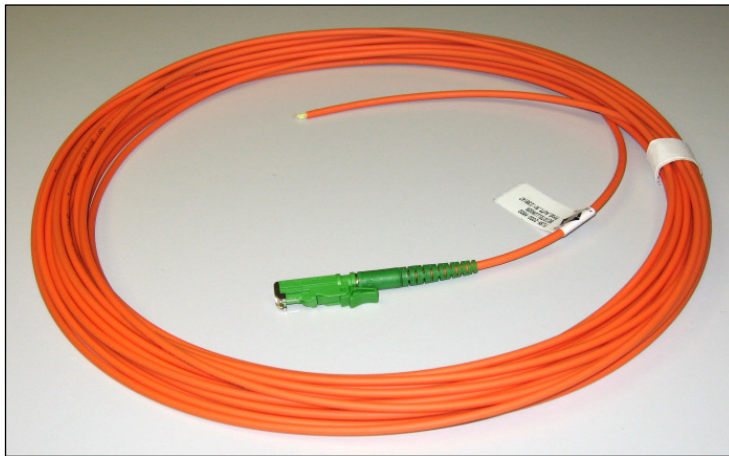
Part number: 28030606

Mechanical characteristics	
Cable stiffness (IEC 60794-1-2, E17A)	1.53 *10 ⁻² Nm ²
Min. bending radius	
without tensile loading (IEC 60794-1-2, E11)	40 mm
with tensile loading (IEC 60794-1-2, E6)	50 mm (500 cycles, 10 N)
Max. tensile strength (IEC 60794-1-2, E1)	
long term	400 N
short term	800 N
Weight	17 kg/km
Max. crush resistance (IEC 60794-1-2, E3)	2000 N/dm (10 min, 3 cycles)
Impact resistance (IEC 60794-1-2, E4)	3 impacts with 1.5 Nm
Impact resistance (extended requirements)	20 impacts with 1.0 Nm (marking stamp 12.5mm & 300mm)
Thermal characteristics	
Operating and storage temperature	- 40°C to + 85°C
Installation temperature	- 5°C to + 50°C
Short term temperature	- 50°C to + 150°C (max. 60 min.)
Flammability	
Flame resistance	IEC 60332-1-2
Circuit Integrity	EN 50200 – PH 30 (30 min, 830 °C, Δα < 3 dB @ 1300 nm)
Smoke density	IEC 61034-2
Halogen free	IEC 60754-2
Fire load	0.31 MJ/m
Standards	
Cable	EN 18700, IEC 60794
Fibre	IEC 60793
Additional Type testing	IEC 60794-1-2, E10, F1, F5, G7; IEC 60794-2-50
Remark	A calibration sheet can be supplied on request.

L10S SIGNLE FIBER CABLE
PIGTAIL, E2000 APC
CONNECTOR

Operations & Maintenance Manual
December 2015

Single Fibre Cable 62.5µm with Connector E2000 APC (Pigtail E2000 62.5µm)
 Part number: 28024020



Specifications	
General description	Single fibre cable 65.5µm with connector E2000 APC on one end.
Cable length	10m
Connector type	E2000, APC, green colour
Connector ferrule	Ceramic, 8° angular polish, premium quality
Lifetime (mating cycles)	Min. 200 cycles
Insertion loss / return loss	Max. 0.3dB / min. 40dB
Pull relief	Min. 100N
Cable design	Metall free, strain relief, aramide yarn reinforcement, self extinguishing, halogen free and low smoke (LSFH), indoor use.
Cable jacket diameter / tube diameter	2.7mm / 900µm
Cable jacket colour / tube colour	Orange / blue
Min. bend radius	50mm during installation (IEC60794-1-2 E11B) 30mm in fixed position
Impact strength	20 impacts @ $W_p=0.74Nm$ / $r=25mm$ (IEC60794-1-2 E4)
Tensile strength	100N during install. and operat., $r>30mm$ (IEC60794-1-2 E1B)
Compressive strength	100N during installation and operation (IEC60794-1-2 E3)
Fibre type	Multimode, graded index
Fibre material	Silica, Ge doped / coating acrylate, transparent
Fibre core / cladding / coating diameter	62.5µm / 125µm / 250µm
Fibre numerical aperture	NA=0.275
Fibre bandwidth-length product	>440MHz*km @ 850nm, >200MHz*km @ 1300nm
Fibre spectral attenuation	<2.8dB/km @ 850nm, <0.6dB/km @ 1300nm
Installation temperature	-5°C – 50°C
Storage / operating temperature	-40°C – 70°C (IEC61300-2-22)
Compliances	Fibre: IEC60793 Cable: IEC60332-1, IEC60332-3 Cat. C (fire propagation) Connector: UL94-V0, IEC61754-15 TIA 604-16

LIOS Technology GmbH
 LINEAR OPTICAL SENSORS
 Schanzenstrasse 39
 Building D9-D13
 51063 Köln/Cologne Germany
 Tel.: +49 221 99887-0
 Fax: +49 221 99887-150
 info@lios-tech.com

Further information about our products and services can be obtained at our internet site www.lios-tech.com

MEMBER OF THE NKT GROUP

This document and all its components is protected by copyright. The recipient shall recognise these rights and shall not disclose the contents, either in whole or in part, of this document without the approval of the publisher to any other third party or use it for any other than its designated purpose.

This Page Left Intentionally Blank

WAGO SYSTEM 750 I/O **SYSTEM & PRO SOFTWARE**

Operations & Maintenance Manual
December 2015

Scalable Industrial Ethernet Solutions

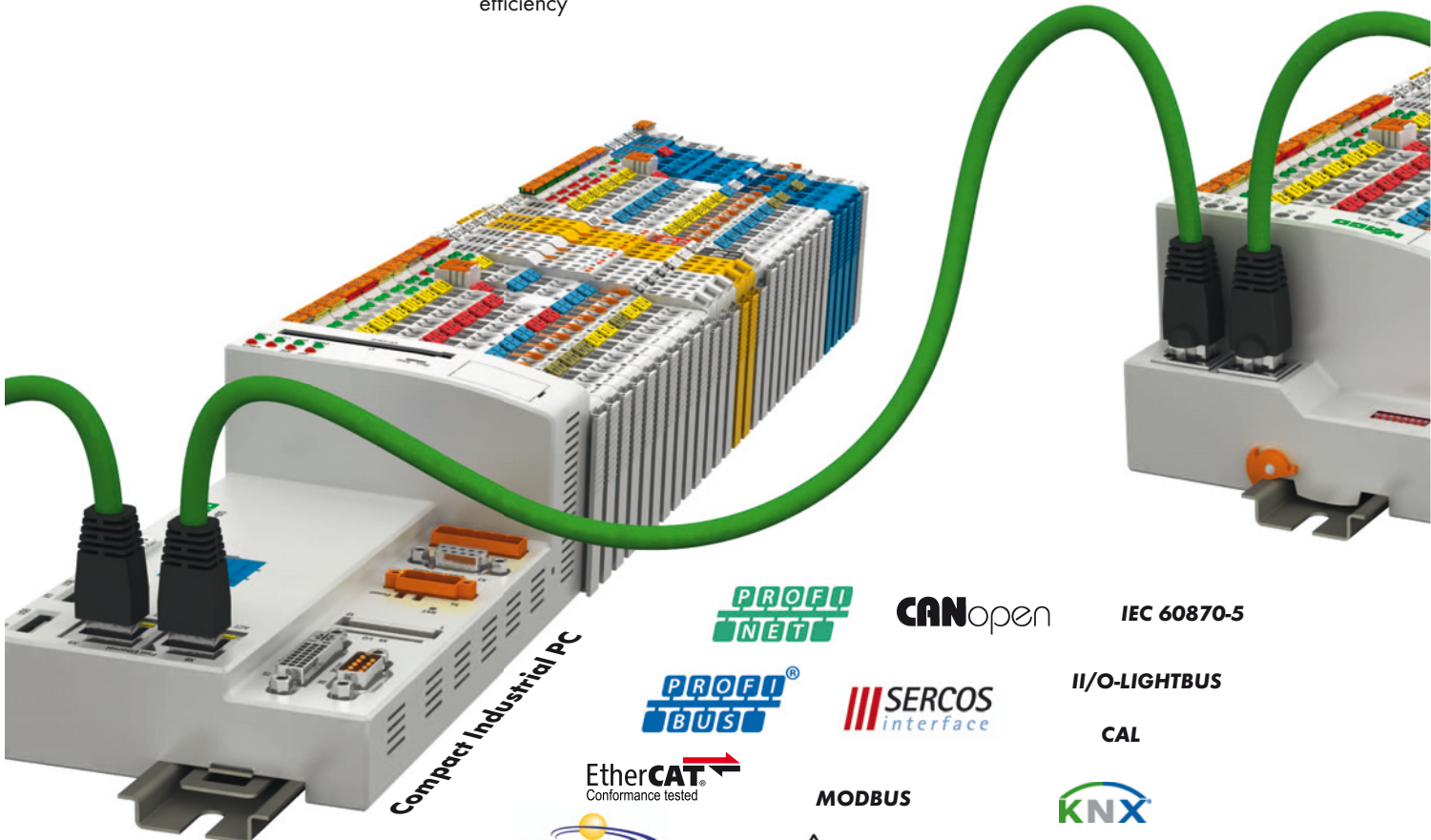


WAGO-I/O-SYSTEM 750, 753 and 758 Series offer future-proof system architecture.

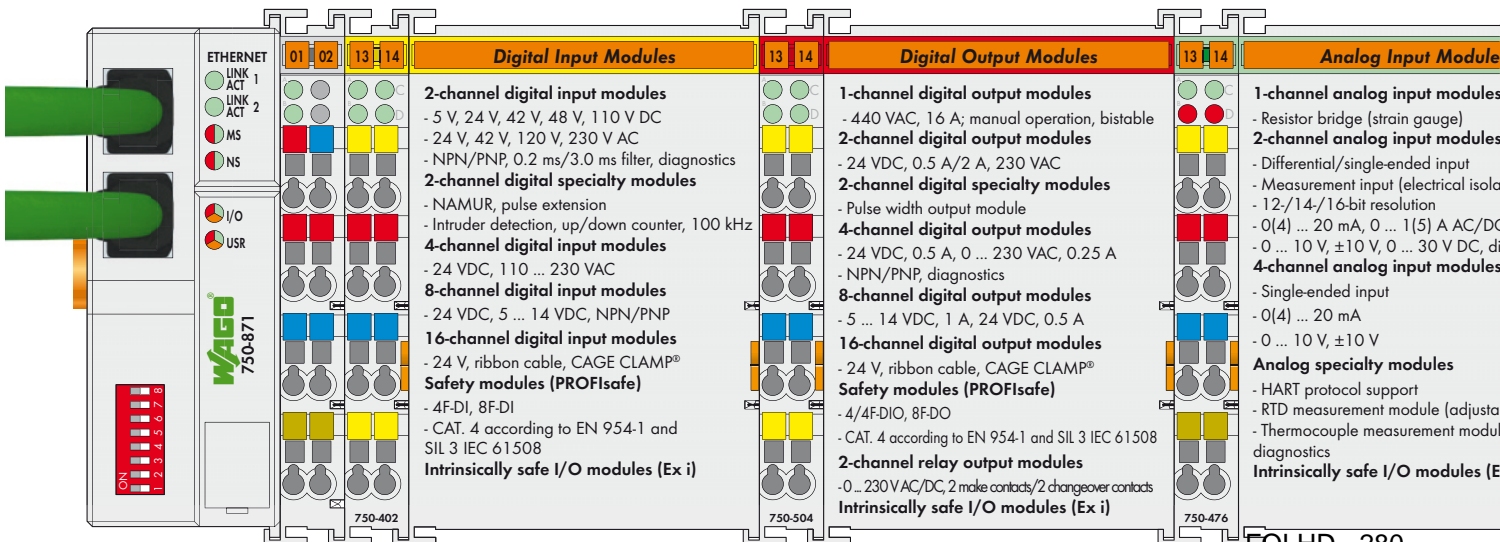
Fine modularity and a fieldbus-independent design are hallmarks of the WAGO-I/O-SYSTEM, which boasts approvals for a diverse range of applications.

- Optimized for process-oriented communication with a scalable performance solution
- High integration density
- Unbeatable price/performance ratio
- The variety of possible applications is almost infinite
- The hardware and system costs, by contrast, have been reduced to a minimum
- For simple use that ensures maximum efficiency

The basic premise of the modular system is reflected in the support of numerous fieldbus systems. Depending on the application, it is possible to choose between fieldbus couplers and controllers for different protocols.



Compact Industrial PC



Optimized for Practical Requirements

Finely modular:

- 1, 2, 4, 8 or 16 channels on an I/O module

Fieldbus-independent:

- Fieldbus couplers & fieldbus controllers for the most common fieldbus protocols & industrial ETHERNET standards

A good investment:

- Fieldbus-independent node design allows easy change to new bus standards while keeping the I/O modules

Clear identification:

- With color-coded group marker carriers and WAGO WSB module identification.

Scalable performance:

- With economy & standard couplers as well as programmable controllers on through to the IPC

High-performance:

- Fieldbus couplers with additional PLC functionality (controllers) for distributed control networks according to IEC 61131-3

Great flexibility:

- Configuration possibilities for digital and analog inputs and outputs and specialty modules with different potentials, capacity, and signals on one fieldbus node

Dependable:

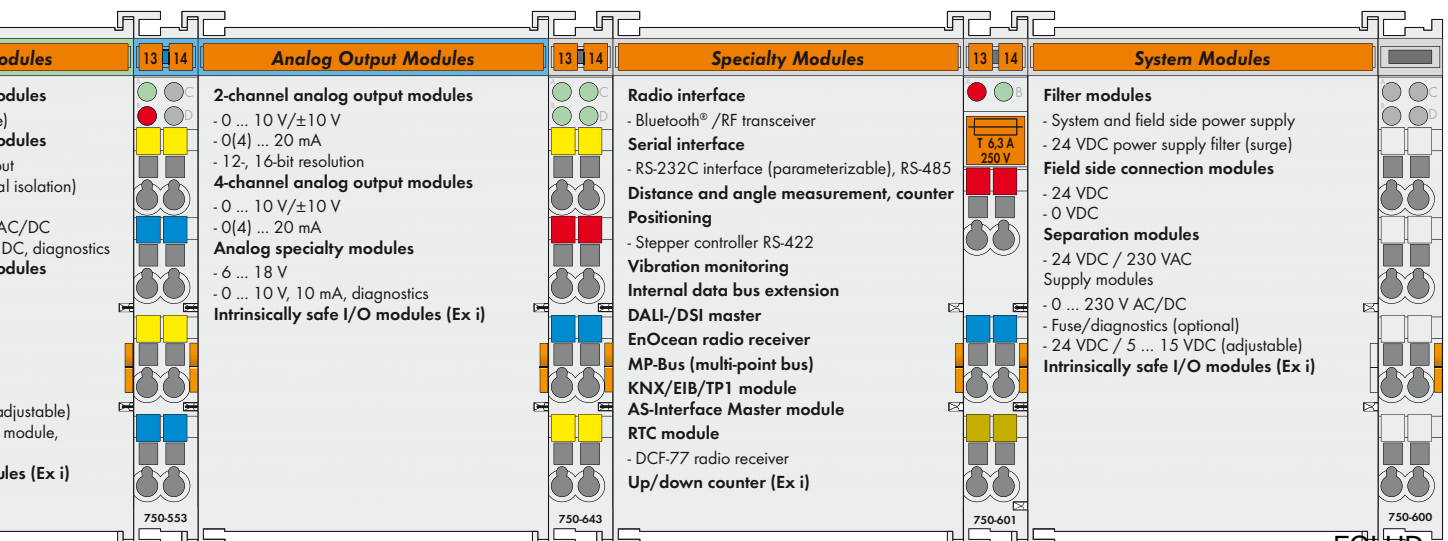
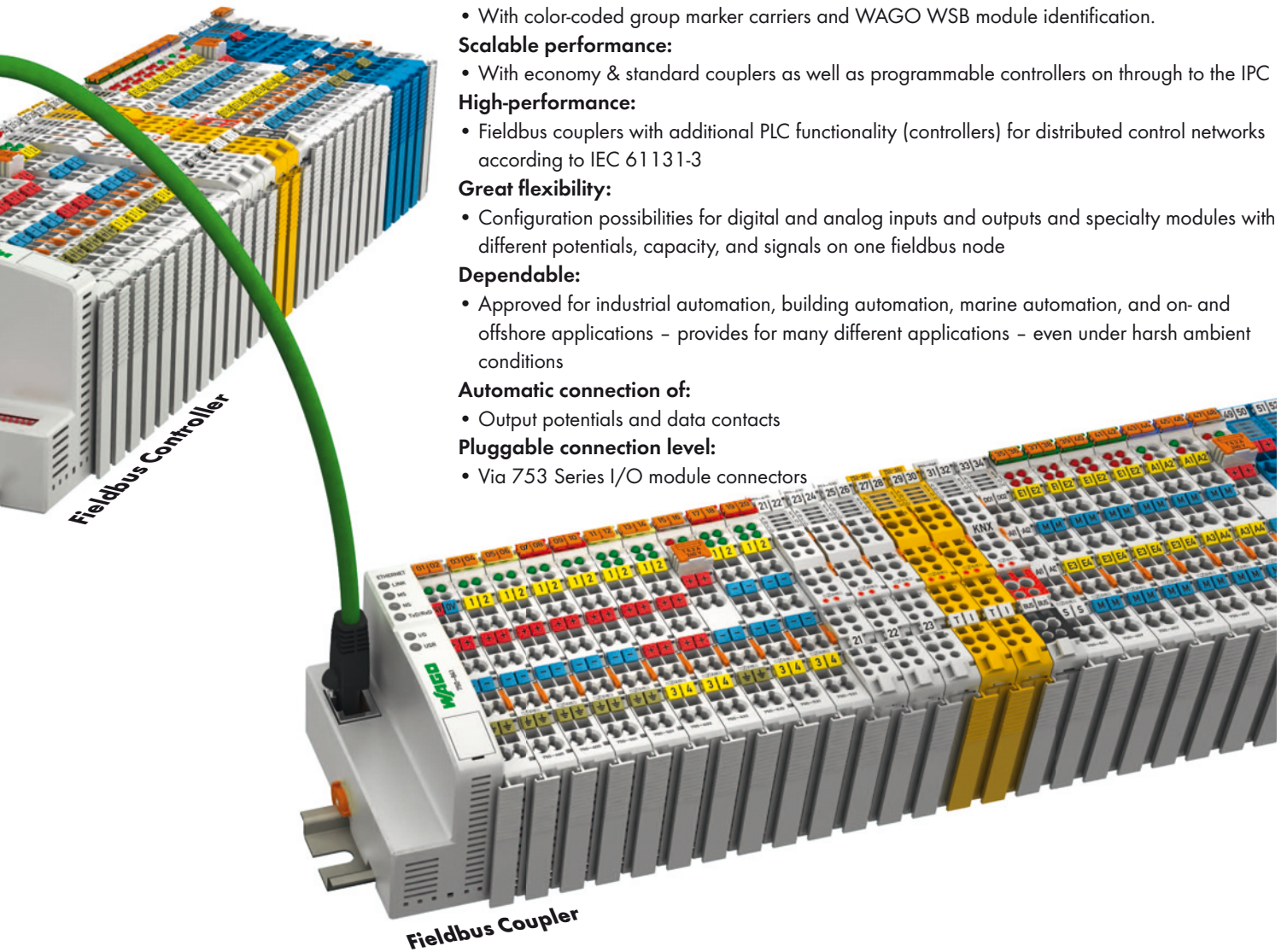
- Approved for industrial automation, building automation, marine automation, and on- and offshore applications – provides for many different applications – even under harsh ambient conditions

Automatic connection of:

- Output potentials and data contacts

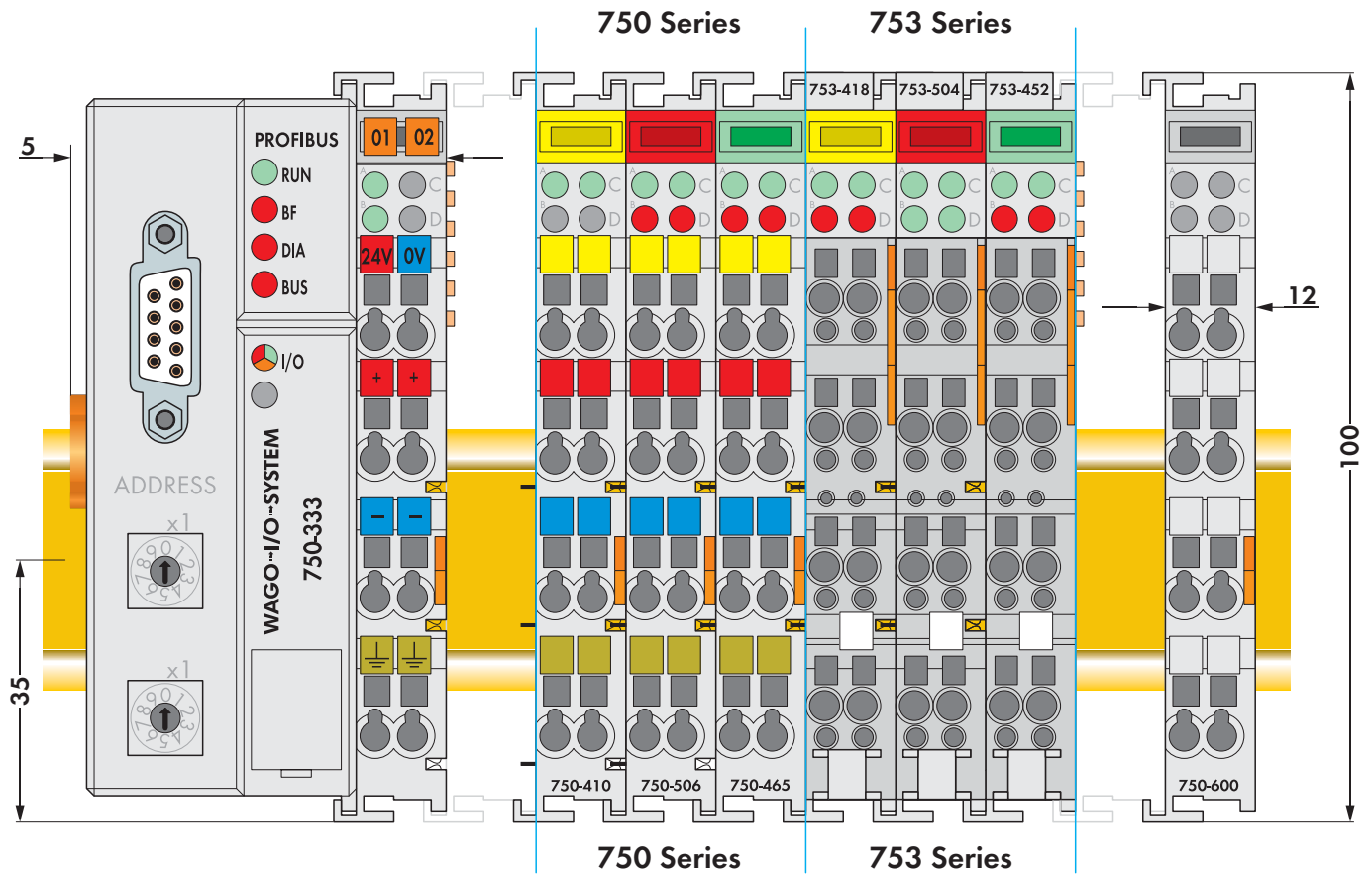
Pluggable connection level:

- Via 753 Series I/O module connectors



I/O System Module Housings, Technical Data

750 and 753 Series



The electronics of the WAGO buscouplers are integrated in a housing with a width of approximately 51 mm.

The electronics of the WAGO I/O System modules are designed into housings ranging from 12 mm up to 48 mm in width, and are known as 750 and 753 Series. 753 Series modules offer the added convenience of pluggable field wiring. 750 and 753 Series modules can co-exist in the same WAGO-I/O-SYSTEM node, and are 100% compatible. Both series offer internal power jumper contacts rated at 10 amps (up to 3 per module), extending field device power to adjacent modules.

Notice:

Depending on the I/O function, all modules do not have the same number of power jumper contacts. Power jumper contacts of adjacent I/O modules must be compatible (i.e. blade contacts must have corresponding spring contacts). Please review the circuit diagrams of the individual modules. An additional power supply module may be necessary.

Note: A bus end module (750-600) is always required at the end of each node and is placed at the end of each node at the end opposite the buscoupler.

Mechanic	
Material	polycarbonate, polyamide 6.6
Dimensions W x H* x L:	
* from upper-edge of DIN rail	
- Coupler/Controller (Standard)	- 51 mm x 65 mm x 100 mm
- Coupler/Controller (ECO)	- 50 mm x 65 mm x 100 mm
- I/O module, single	- 12 mm x 64 mm x 100 mm
- I/O module, double	- 24 mm x 64 mm x 100 mm
- I/O module, fourfold	- 48 mm x 64 mm x 100 mm
Installation	on DIN 35 rail with interlock
Modular by	double featherkey-dovetail
Mounting position	any position
Marking	Standard markers and 8 x 47 mm markers for group marker carriers
Connection	
Connection type	CAGE CLAMP®
Wire range	0.08 mm ² ... 2.5 mm ² , AWG 28 ... 14
Stripped length	8 ... 9 mm, 9 ... 10 mm for components with pluggable wiring (753-xxx)
Contacts	
Power jumper contacts	blade/spring contact, self-cleaning
Current I _{max}	10 A
Voltage drop (I=10 A)	< 1 V/64 modules
Data contacts	slide contact, hard gold plated 1.5 μ, self-cleaning
General Environmental Conditions	
Operating temperature	0 °C ... +55 °C,
for components with extended temperature range (750-xxx/025-xxx)	-20 °C ... +60 °C
Storage temperature	-25 °C ... +85 °C
for components with extended temperature range (750-xxx/025-xxx)	-40 °C ... +85 °C
Relative air humidity	95 % without condensation
Resistance to harmful substances	acc. to IEC 60068-2-42 and IEC 60068-2-43
Maximum pollutant concentration at relative humidity < 75%	SO ₂ ≤ 25 ppm H ₂ S ≤ 10 ppm
Vibration resistance	acc. to IEC 60068-2-6
Shock resistance	acc. to IEC 60068-2-27
Pollution degree II	acc. to IEC 61131-2
Air and creepage distance	acc. to IEC 60664-1
Degree of protection	IP20
Length of entire node	≤ 831 mm

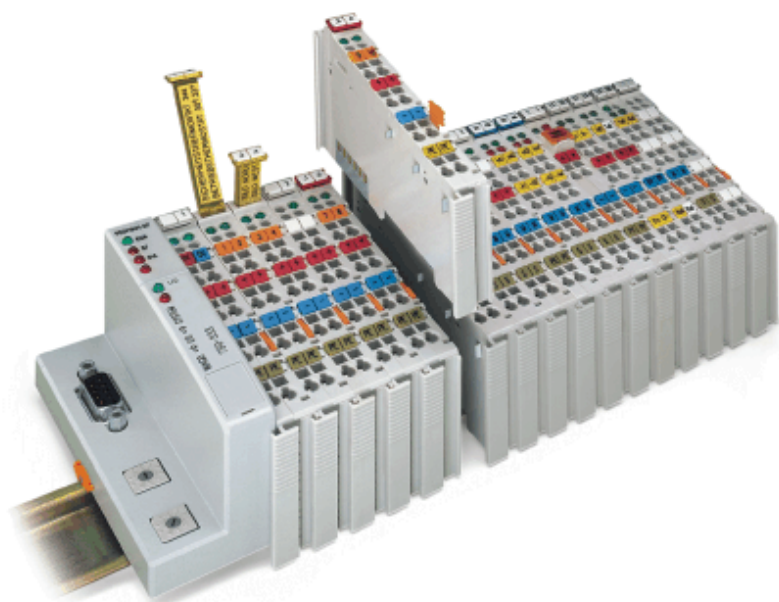
Without taking protective measures the WAGO-I/O-SYSTEM 750 and 753 must not be used:

- in places with unfavorable conditions, for example with dust, corrosive vapors or gases
- in places with high ionizing radiation

This Page Left Intentionally Blank

WAGO I/O SYSTEM 750

Modular I/O System



Design Notes

Guidlines and Recommendations
for Increasing Operational Safety

Version 2.0.0

Copyright © 2005 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0

Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55

Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the correctness and completeness of this documentation. However, as errors can never be fully excluded, we would appreciate any information or ideas at any time.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally trademark or patent protected.

TABLE OF CONTENTS

1 Important Comments.....	4
1.1 Legal Principles.....	4
1.1.1 Copyright.....	4
1.1.2 Personnel Qualification.....	4
1.1.3 Intended Use.....	4
1.2 Symbols.....	5
1.3 Font Conventions.....	5
1.4 Number Notation.....	5
1.5 Safety Notes.....	6
1.6 Scope of Validity.....	7
2 The WAGO-I/O-SYSTEM 750.....	8
2.1 System Description.....	8
2.2 Technical Data.....	9
2.3 Manufacturing Number.....	15
2.4 Component Update.....	16
2.5 Storage, Assembly and Transport.....	16
2.6 Mechanical Setup.....	17
2.6.1 Installation Position.....	17
2.6.2 Total Expansion.....	17
2.6.3 Assembly onto Carrier Rail.....	18
2.6.4 Spacing.....	19
2.6.5 Plugging and Removal of the Components.....	20
2.6.6 Assembly Sequence.....	21
2.6.7 Internal Bus/Data Contacts.....	22
2.6.8 Power Contacts.....	23
2.6.9 Wire connection.....	24
2.7 Power Supply.....	25
2.7.1 Isolation.....	25
2.7.2 System Supply.....	26
2.7.3 Field Supply.....	29
2.7.4 Supplementary power supply regulations.....	33
2.7.5 Supply example.....	34
2.7.6 Power Supply Unit.....	35
2.8 Grounding.....	36
2.8.1 Grounding the DIN Rail.....	36
2.8.2 Grounding Function.....	37
2.8.3 Grounding Protection.....	38
2.9 Shielding (Screening).....	39
2.9.1 General.....	39
2.9.2 Bus Conductors.....	39
2.9.3 Signal Conductors.....	39
2.9.4 WAGO Shield (Screen) Connecting System.....	40
2.10 Assembly Guidelines/Standards.....	40
3 Check list (Draft).....	41

1 Important Comments

To ensure fast installation and start-up of the units described in this manual, we strongly recommend that the following information and explanations are carefully read and abided by.

1.1 Legal Principles

1.1.1 Copyright

This manual is copyrighted, together with all figures and illustrations contained therein. Any use of this manual which infringes the copyright provisions stipulated herein, is not permitted. Reproduction, translation and electronic and photo-technical archiving and amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG. Non-observance will entail the right of claims for damages.

WAGO Kontakttechnik GmbH & Co. KG reserves the right to perform modifications allowed by technical progress. In case of grant of a patent or legal protection of utility patents all rights are reserved by WAGO Kontakttechnik GmbH & Co. KG. Products of other manufacturers are always named without referring to patent rights. The existence of such rights can therefore not be ruled out.

1.1.2 Personnel Qualification

The use of the product detailed in this manual is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the valid standards. WAGO Kontakttechnik GmbH & Co. KG declines all liability resulting from improper action and damage to WAGO products and third party products due to non-observance of the information contained in this manual.

1.1.3 Intended Use

For each individual application, the components supplied are to work with a dedicated hardware and software configuration. Modifications are only permitted within the framework of the possibilities documented in the manuals. All other changes to the hardware and/or software and the non-conforming use of the components entail the exclusion of liability on part of WAGO Kontakttechnik GmbH & Co. KG.

Please direct any requirements pertaining to a modified and/or new hardware or software configuration directly to WAGO Kontakttechnik GmbH & Co. KG.

1.2 Symbols



Danger

Always abide by this information to protect persons from injury.



Warning

Always abide by this information to prevent damage to the device.



Attention

Marginal conditions must always be observed to ensure smooth operation.



ESD (Electrostatic Discharge)

Warning of damage to the components by electrostatic discharge. Observe the precautionary measure for handling components at risk.



Note

Routines or advice for efficient use of the device and software optimization.



More information

References on additional literature, manuals, data sheets and INTERNET pages

1.3 Font Conventions

<i>Italic</i>	Names of path and files are marked italic i.e.: <i>C:\programs\WAGO-IO-CHECK</i>
<i>Italic</i>	Menu items are marked as bold italic i.e.: <i>Save</i>
\	A backslash between two names marks a sequence of menu items i.e.: <i>FileNew</i>
END	Press buttons are marked as bold with small capitals i.e.: ENTER
<>	Keys are marked bold within angle brackets i.e.: <F5>
Courier	Program code is printed with the font Courier. i.e.: END_VAR

1.4 Number Notation

Number Code	Example	Note
Decimal	100	normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	Within ', Nibble separated with dots

1.5 Safety Notes



Attention

Switch off the system prior to working on bus modules!

In the event of deformed contacts, the module in question is to be replaced, as its functionality can no longer be ensured on a long-term basis.

The components are not resistant against materials having seeping and insulating properties. Belonging to this group of materials is: e.g. aerosols, silicones, triglycerides (found in some hand creams).

If it cannot be ruled out that these materials appear in the component environment, then additional measures are to be taken:

- installation of the components into an appropriate enclosure
 - handling of the components only with clean tools and materials.
-



Attention

Cleaning of soiled contacts may only be done with ethyl alcohol and leather cloths. Thereby, the ESD information is to be regarded.

Do not use any contact spray. The spray may impair the functioning of the contact area.

The WAGO-I/O-SYSTEM 750 and its components are an open system. It must only be assembled in housings, cabinets or in electrical operation rooms. Access must only be given via a key or tool to authorized qualified personnel.

The relevant valid and applicable standards and guidelines concerning the installation of switch boxes are to be observed.



ESD (Electrostatic Discharge)

The modules are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the modules, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, e.g. gold contacts.

1.6 Scope of Validity

The project plan information for the WAGO-I/O-SYSTEM 750 provides recommendations for the use of the modular I/O system. Guidelines and tips for enhancing operational safety are provided therein.

This document is only valid in connection with the manuals and the current WAGO catalogue.

Up-to-date information, manuals and catalog data can be found on the Internet under: www.wago.com.

The setup guidelines of the relevant Fieldbus organization are to be observed. Relevant information for the I/O System in excerpts is documented in the manuals on the WAGO- I/O-SYSTEM 750.

2 The WAGO-I/O-SYSTEM 750

2.1 System Description

The WAGO-I/O-SYSTEM 750 is a modular, fieldbus independent I/O system. It is comprised of a fieldbus coupler/controller (1) and connected fieldbus modules (2) for any type of signal. Together, these make up the fieldbus node. The end module (3) completes the node.

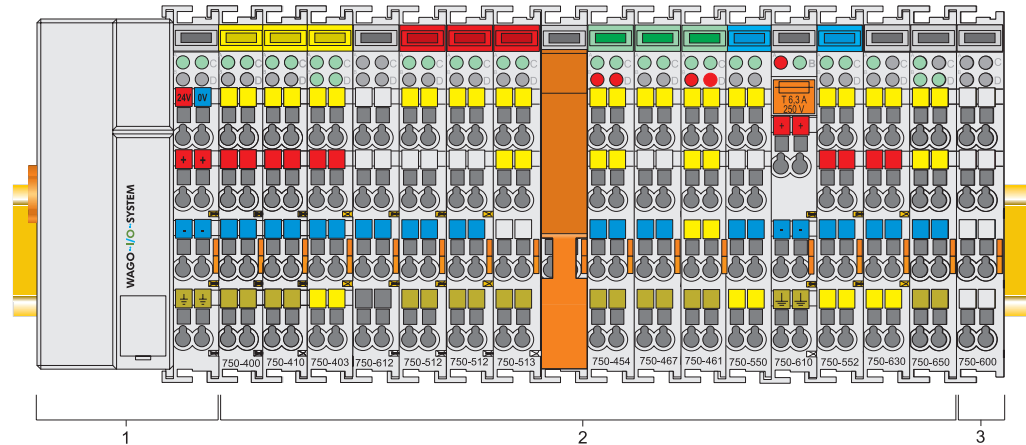


Fig. 2-1: Fieldbus node

g0xxx00x

Couplers/controllers for fieldbus systems such as PROFIBUS, INTERBUS, ETHERNET TCP/IP, CAN (CANopen, DeviceNet, CAL), MODBUS, LON and others are available.

The coupler/controller contains the fieldbus interface, electronics and a power supply terminal. The fieldbus interface forms the physical interface to the relevant fieldbus. The electronics process the data of the bus modules and make it available for the fieldbus communication. The 24 V system supply and the 24 V field supply are fed in via the integrated power supply terminal. The fieldbus coupler communicates via the relevant fieldbus. The programmable fieldbus controller (PFC) enables the implementation of additional PLC functions. Programming is done with the WAGO-I/O-PRO 32 in accordance with IEC 61131-3.

Bus modules for diverse digital and analog I/O functions as well as special functions can be connected to the coupler/controller. The communication between the coupler/controller and the bus modules is carried out via an internal bus.

The WAGO-I/O-SYSTEM 750 has a clear port level with LEDs for status indication, insertable mini WSB markers and pullout group marker carriers. The 3-wire technology supplemented by a ground wire connection allows for direct sensor/actuator wiring.

2.2 Technical Data

Mechanic	
Material	Polycarbonate, Polyamide 6.6
Dimensions W x H* x L * from upper edge of DIN 35 rail	
- Coupler/Controller (Standard)	- 51 mm x 65 mm x 100 mm
- Coupler/Controller (ECO)	- 50 mm x 65 mm x 100 mm
- Coupler/Controller (FireWire)	- 62 mm x 65 mm x 100 mm
- I/O module, single	- 12 mm x 64 mm x 100 mm
- I/O module, double	- 24 mm x 64 mm x 100 mm
- I/O module, fourfold	- 48 mm x 64 mm x 100 mm
Installation	on DIN 35 with interlock
modular by	double featherkey-dovetail
Mounting position	any position
Marking	marking label type 247 and 248 paper marking label 8 x 47 mm
Connection	
Connection type	CAGE CLAMP®
Wire range	0.08 mm ² ... 2.5 mm ² , AWG 28-14
Stripped length	8 – 9 mm, 9 – 10 mm for components with pluggable wiring (753-xxx)
Contacts	
Power jumpers contacts	blade/spring contact self-cleaning
Current via power contacts _{max}	10 A
Voltage drop at I _{max}	< 1 V/64 modules
Data contacts	slide contact, hard gold plated 1.5 µm, self-cleaning
Climatic environmental conditions	
Operating temperature	0 °C ... 55 °C, -20 °C ... +60 °C for components with extended temperature range (750-xxx/025-xxx)
Storage temperature	-20 °C ... +85 °C
Relative humidity	5 % to 95 % without condensation
Resistance to harmful substances	acc. to IEC 60068-2-42 and IEC 60068-2-43
Maximum pollutant concentration at relative humidity < 75%	SO ₂ ≤ 25 ppm H ₂ S ≤ 10 ppm
Special conditions	Ensure that additional measures for components are taken, which are used in an environment involving: – dust, caustic vapors or gasses – ionization radiation.

Safe electrical isolation				
Air and creepage distance	acc. to IEC 60664-1			
Degree of pollution acc. To IEC 61131-2	2			
Degree of protection				
Degree of protection	IP 20			
Electromagnetic compatibility				
Immunity to interference for industrial areas acc. to EN 61000-6-2 (2001)				
Test specification	Test values	Strength class	Evaluation criteria	
EN 61000-4-2 ESD	4 kV/8 kV (contact/air)	2/3	B	
EN 61000-4-3 electromagnetic fields	10 V/m 80 MHz ... 1 GHz	3	A	
EN 61000-4-4 burst	1 kV/2 kV (data/supply)	2/3	B	
EN 61000-4-5 surge	Data:	-/- (line/line)	B	
		1 kV (line/earth)		2
	DC supply:	0.5 kV (line/line)	1	B
		0.5 kV (line/earth)	1	
	AC supply:	1 kV (line/line)	2	B
		2 kV (line/earth)	3	
EN 61000-4-6 RF disturbances	10 V/m 80 % AM (0.15 ... 80 MHz)	3	A	
Emission of interference for industrial areas acc. to EN 61000-6-4 (2001)				
Test specification	Limit values/[QP]*	Frequency range	Distance	
EN 55011 (AC supply, conducted)	79 dB (µV)	150 kHz ... 500 kHz		
	73 dB (µV)	500 kHz ... 30 MHz		
EN 55011 (radiated)	40 dB (µV/m)	30 MHz ... 230 MHz	10 m	
	47 dB (µV/m)	230 MHz ... 1 GHz	10 m	
Emission of interference for residential areas acc. to EN 61000-6-3 (2001)				
Test specification	Limit values/[QP]*	Frequency range	Distance	
EN 55022 (AC supply, conducted)	66 ... 56 dB (µV)	150 kHz ... 500 kHz		
	56 dB (µV)	500 kHz ... 5 MHz		
	60 dB (µV)	5 MHz ... 30 MHz		
EN 55022 (DC supply/data, conducted)	40 ... 30 dB (µA)	150 kHz ... 500 kHz		
	30 dB (µA)	500 kHz ... 30 MHz		
EN 55022 (radiated)	30 dB (µV/m)	30 MHz ... 230 MHz	10 m	
	37 dB (µV/m)	230 MHz ... 1 GHz	10 m	

Mechanical strength acc. to IEC 61131-2		
Test specification	Frequency range	Limit value
IEC 60068-2-6 vibration	$5 \text{ Hz} \leq f < 9 \text{ Hz}$	1.75 mm amplitude (permanent) 3.5 mm amplitude (short term)
	$9 \text{ Hz} \leq f < 150 \text{ Hz}$	0.5 g (permanent) 1 g (short term)
	Note on vibration test: a) Frequency change: max. 1 octave/minute b) Vibration direction: 3 axes	
IEC 60068-2-27 shock		15 g
	Note on shock test: a) Type of shock: half sine b) Shock duration: 11 ms c) Shock direction: 3x in positive and 3x in negative direction for each of the three mutually perpendicular axes of the test specimen	
IEC 60068-2-32 free fall		1 m (module in original packing)

*) QP: Quasi Peak



Note:

If the technical data of components differ from the values described here, the technical data shown in the manuals of the respective components shall be valid.

For Products of the WAGO-I/O-SYSTEM 750 with ship specific approvals, supplementary guidelines are valid:

Electromagnetic compatibility				
Immunity to interference acc. to Germanischer Lloyd (2003)				
Test specification	Test values		Strength class	
			Evaluation criteria	
IEC 61000-4-2 ESD	6 kV/8 kV (contact/air)		3/3	B
IEC 61000-4-3 electromagnetic fields	10 V/m 80 MHz ... 2 GHz		3	A
IEC 61000-4-4 burst	1 kV/2 kV (data/supply)		2/3	A
IEC 61000-4-5 surge	AC/DC Supply:	0.5 kV (line/line)	1	A
		1 kV (line/earth)	2	
IEC 61000-4-6 RF disturbances	10 V/m 80 % AM (0.15 ... 80 MHz)		3	A
Type test AF disturbances (harmonic waves)	3 V, 2 W		-	A
Type test high voltage	755 V DC 1500 V AC		-	-
Emission of interference acc. to Germanischer Lloyd (2003)				
Test specification	Limit values	Frequency range	Distance	
Type test (EMC1, conducted) allows for ship bridge control applications	96 ... 50 dB (µV)	10 kHz ... 150 kHz		
	60 ... 50 dB (µV)	150 kHz ... 350 kHz		
	50 dB (µV)	350 kHz ... 30 MHz		
Type test (EMC1, radiated) allows for ship bridge control applications außer für:	80 ... 52 dB (µV/m)	150 kHz ... 300 kHz	3 m	
	52 ... 34 dB (µV/m)	300 kHz ... 30 MHz	3 m	
	54 dB (µV/m)	30 MHz ... 2 GHz	3 m	
	24 dB (µV/m)	156 MHz ... 165 MHz	3 m	
Mechanical strength acc. to Germanischer Lloyd (2003)				
Test specification	Frequency range	Limit value		
IEC 60068-2-6 vibration (category A – D)	2 Hz ≤ f < 25 Hz	± 1.6 mm amplitude (permanent)		
	25 Hz ≤ f < 100 Hz	4 g (permanent)		
	Note on vibration test: a) Frequency change: max. 1 octave/minute b) Vibration direction: 3 axes			

Range of application	Required specification emission of interference	Required specification immunity to interference
Industrial areas	EN 61000-6-4 (2001)	EN 61000-6-2 (2001)
Residential areas	EN 61000-6-3 (2001)*)	EN 61000-6-1 (2001)

*) The system meets the requirements on emission of interference in residential areas with the fieldbus coupler/controller for:

ETHERNET 750-342/-841/-842

LonWorks 750-319/-819

CANopen 750-337/-837

DeviceNet 750-306/-806

MODBUS 750-312/-314/ -315/ -316
750-812/-814/ -815/ -816

With a special permit, the system can also be implemented with other fieldbus couplers/controllers in residential areas (housing, commercial and business areas, small-scale enterprises). The special permit can be obtained from an authority or inspection office. In Germany, the Federal Office for Post and Telecommunications and its branch offices issues the permit.

It is possible to use other field bus couplers/controllers under certain boundary conditions. Please contact WAGO Kontakttechnik GmbH & Co. KG.

Maximum power dissipation of the components	
Bus modules	0.8 W / bus terminal (total power dissipation, system/field)
Fieldbus coupler/controller	2.0 W / coupler/controller



Warning

The power dissipation of all installed components must not exceed the maximum conductible power of the housing (cabinet).

When dimensioning the housing, care is to be taken that even under high external temperatures, the temperature inside the housing does not exceed the permissible ambient temperature of 55 °C.

Dimensions

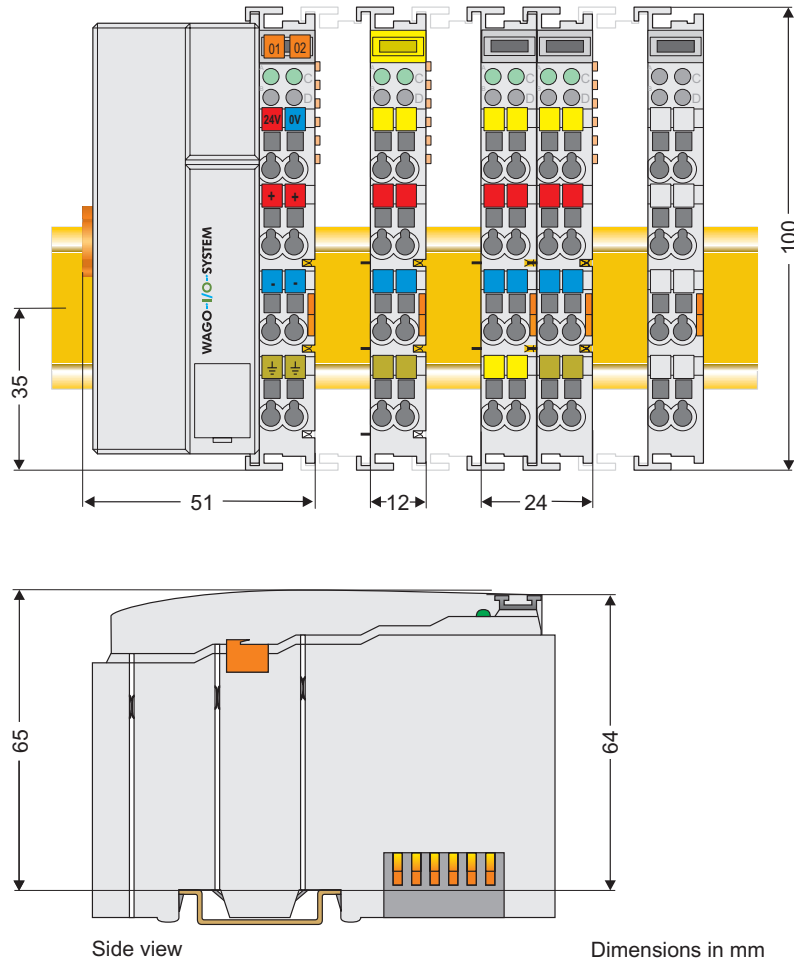


Fig. 2-2: Dimensions

g01xx05e



Note:

The illustration shows a standard coupler. For detailed dimensions, please refer to the technical data of the respective coupler/controller.

2.3 Manufacturing Number

The manufacturing number indicates the delivery status directly after production.

This number is part of the lateral marking on the component.

In addition, starting from calendar week 43/2000 the manufacturing number is also printed on the cover of the configuration and programming interface of the fieldbus coupler or controller.

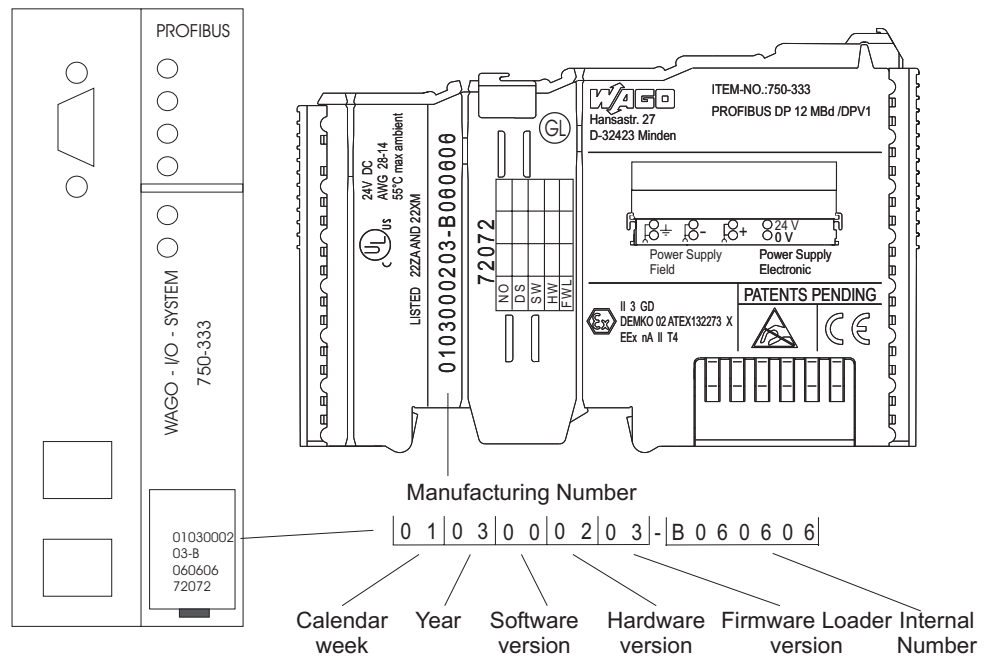


Fig. 2-3: Example: Manufacturing Number of a PROFIBUS fieldbus coupler 750-333

g01xx15e

The manufacturing number consists of the production week and year, the software version (if available), the hardware version of the component, the firmware loader (if available) and further internal information for WAGO Kontakttechnik GmbH.

2.4 Component Update

For the case of an Update of one component, the lateral marking on each component contains a prepared matrix.

This matrix makes columns available for altogether three updates to the entry of the current update data, like production order number (NO; starting from calendar week 13/2004), update date (DS), software version (SW), hardware version (HW) and the firmware loader version (FWL, if available).

Update Matrix

Current Version data for:	1. Update	2. Update	3. Update	
Production Order Number	NO			<- Only starting from Calendar week 13/2004
Datestamp	DS			
Software index	SW			
Hardware index	HW			
Firmware loader index	FWL			<- Only for coupler/controller

If the update of a component took place, the current version data are registered into the columns of the matrix.

Additionally with the update of a fieldbus coupler or controller also the cover of the configuration and programming interface of the coupler or controller is printed on with the current manufacturing and production order number.

The original manufacturing data on the housing of the component remain thereby.

2.5 Storage, Assembly and Transport

Wherever possible, the components are to be stored in their original packaging. Likewise, the original packaging provides optimal protection during transport.

When assembling or repacking the components, the contacts must not be soiled or damaged. The components must be stored and transported in appropriate containers/packaging. Thereby, the ESD information is to be regarded.

Statically shielded transport bags with metal coatings are to be used for the transport of open components for which soiling with amine, amide and silicone has been ruled out, e.g. 3M 1900E.

2.6 Mechanical Setup

2.6.1 Installation Position

Along with horizontal and vertical installation, all other installation positions are allowed.



Attention

In the case of vertical assembly, an end stop has to be mounted as an additional safeguard against slipping.

WAGO item 249-116 End stop for DIN 35 rail, 6 mm wide

WAGO item 249-117 End stop for DIN 35 rail, 10 mm wide

2.6.2 Total Expansion

The length of the module assembly (including one end module of 12mm width) that can be connected to the coupler/controller is 780mm. When assembled, the I/O modules have a maximum length of 768mm.

Examples:

- 64 I/O modules of 12mm width can be connected to one coupler/controller.
- 32 I/O modules of 24mm width can be connected to one coupler/controller.

Exception:

The number of connected I/O modules also depends on which type of coupler/controller is used. For example, the maximum number of I/O modules that can be connected to a Profibus coupler/controller is 63 without end module. The maximum total expansion of a node is calculated as follows:



Warning

The maximum total length of a node without coupler/controller must not exceed 780mm. Furthermore, restrictions made on certain types of couplers/controllers must be observed (e.g. for Profibus).

2.6.3 Assembly onto Carrier Rail

2.6.3.1 Carrier rail properties

All system components can be snapped directly onto a carrier rail in accordance with the European standard EN 50022 (DIN 35).



Warning

WAGO supplies standardized carrier rails that are optimal for use with the I/O system. If other carrier rails are used, then a technical inspection and approval of the rail by WAGO Kontakttechnik GmbH should take place.

Carrier rails have different mechanical and electrical properties. For the optimal system setup on a carrier rail, certain guidelines must be observed:

- The material must be non-corrosive.
- Most components have a contact to the carrier rail to ground electromagnetic disturbances. In order to avoid corrosion, this tin-plated carrier rail contact must not form a galvanic cell with the material of the carrier rail which generates a differential voltage above 0.5 V (saline solution of 0.3% at 20°C).
- The carrier rail must optimally support the EMC measures integrated into the system and the shielding of the bus module connections.
- A sufficiently stable carrier rail should be selected and, if necessary, several mounting points (every 20 cm) should be used in order to prevent bending and twisting (torsion).
- The geometry of the carrier rail must not be altered in order to secure the safe hold of the components. In particular, when shortening or mounting the carrier rail, it must not be crushed or bent.
- The base of the I/O components extends into the profile of the carrier rail. For carrier rails with a height of 7.5 mm, mounting points are to be riveted under the node in the carrier rail (slotted head captive screws or blind rivets).

2.6.3.2 WAGO DIN Rail

WAGO carrier rails meet the electrical and mechanical requirements.

Item Number	Description
210-113 /-112	35 x 7.5; 1 mm; steel yellow chromated; slotted/unslotted
210-114 /-197	35 x 15; 1.5 mm; steel yellow chromated; slotted/unslotted
210-118	35 x 15; 2.3 mm; steel yellow chromated; unslotted
210-198	35 x 15; 2.3 mm; copper; unslotted
210-196	35 x 7.5; 1 mm; aluminum; unslotted

2.6.4 Spacing

The spacing between adjacent components, cable conduits, casing and frame sides must be maintained for the complete field bus node.

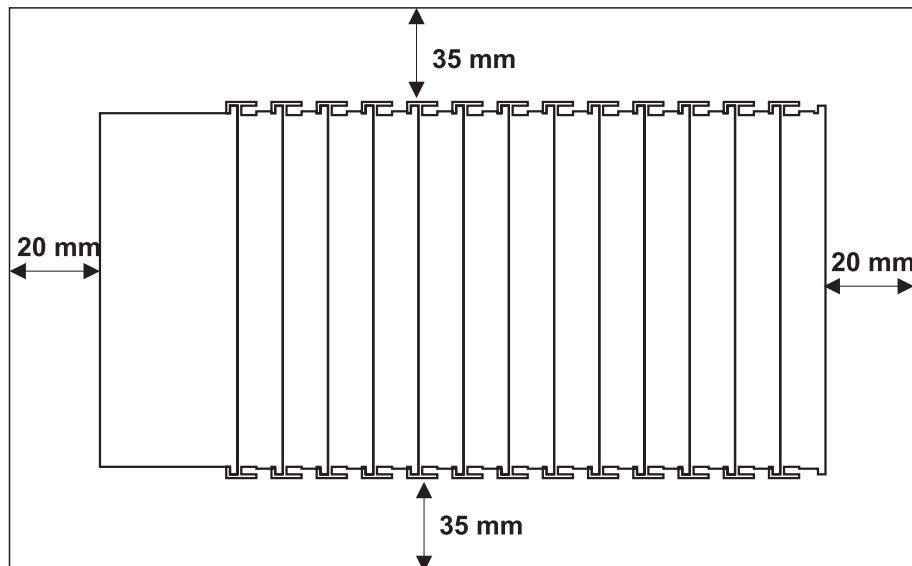


Fig. 2-4: Spacing

g01xx13x

The spacing creates room for heat transfer, installation or wiring. The spacing to cable conduits also prevents conducted electromagnetic interferences from influencing the operation.

2.6.5 Plugging and Removal of the Components



Warning

Before work is done on the components, the voltage supply must be turned off.

In order to safeguard the coupler/controller from jamming, it should be fixed onto the carrier rail with the locking disc. To do so, push on the upper groove of the locking disc using a screwdriver.

To pull out the fieldbus coupler/controller, release the locking disc by pressing on the bottom groove with a screwdriver and then pulling the orange colored unlocking lug.

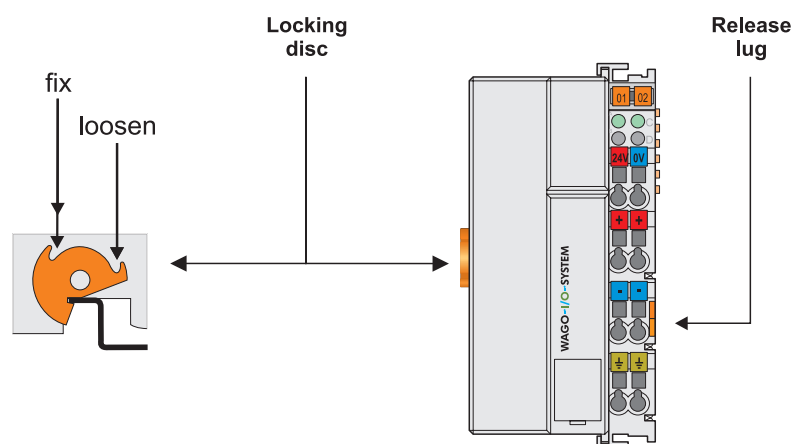


Fig. 2-5: Coupler/Controller and unlocking lug

g01xx12e

It is also possible to release an individual I/O module from the unit by pulling an unlocking lug.

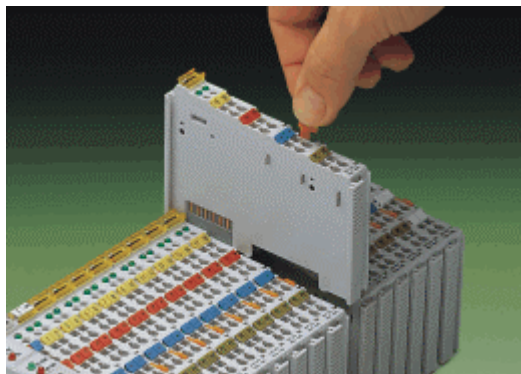


Fig. 2-6: removing bus terminal

p0xxx01x



Danger

Ensure that an interruption of the PE will not result in a condition which could endanger a person or equipment!

For planning the ring feeding of the ground wire, please see chapter 2.6.3.

2.6.6 Assembly Sequence

All system components can be snapped directly on a carrier rail in accordance with the European standard EN 50022 (DIN 35).

The reliable positioning and connection is made using a tongue and groove system. Due to the automatic locking, the individual components are securely seated on the rail after installing.

Starting with the coupler/controller, the bus modules are assembled adjacent to each other according to the project planning. Errors in the planning of the node in terms of the potential groups (connection via the power contacts) are recognized, as the bus modules with power contacts (male contacts) cannot be linked to bus modules with fewer power contacts.



Attention

Always link the bus modules with the coupler/controller, and always plug from above.



Warning

Never plug bus modules from the direction of the end terminal. A ground wire power contact, which is inserted into a terminal without contacts, e.g. a 4-channel digital input module, has a decreased air and creepage distance to the neighboring contact in the example DI4.

Always terminate the fieldbus node with an end module (750-600).

2.6.7 Internal Bus/Data Contacts

Communication between the coupler/controller and the bus modules as well as the system supply of the bus modules is carried out via the internal bus. It is comprised of 6 data contacts, which are available as self-cleaning gold spring contacts.

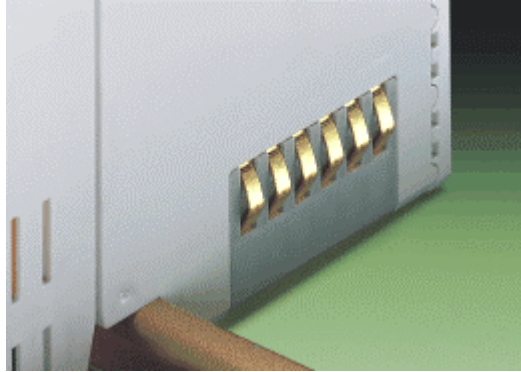


Fig. 2-7: Data contacts

p0xxx07x



Warning

Do not touch the gold spring contacts on the I/O modules in order to avoid soiling or scratching!



ESD (Electrostatic Discharge)

The modules are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the modules, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, e.g. gold contacts.

2.6.8 Power Contacts

Self-cleaning power contacts, are situated on the side of the components which further conduct the supply voltage for the field side. These contacts come as touchproof spring contacts on the right side of the coupler/controller and the bus module. As fitting counterparts the module has male contacts on the left side.



Danger

The power contacts are sharp-edged. Handle the module carefully to prevent injury.



Attention

Please take into consideration that some bus modules have no or only a few power jumper contacts. The design of some modules does not allow them to be physically assembled in rows, as the grooves for the male contacts are closed at the top.

Power jumper contacts

Blade	0	0	3	2
Spring	0	3	3	2

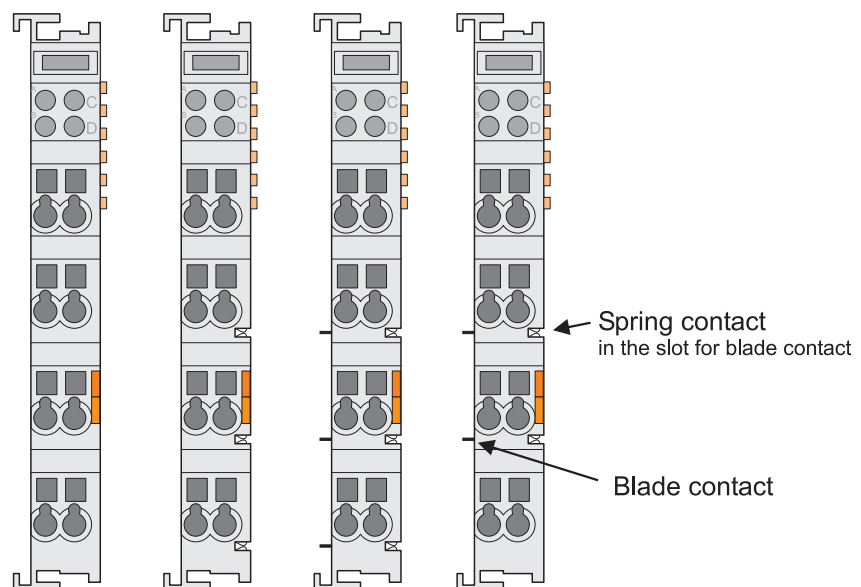


Fig. 2-8: Example for the arrangement of power contacts

g0xxx05e

Recommendation

With the WAGO ProServe® Software smartDESIGNER, the assembly of a fieldbus node can be configured. The configuration can be tested via the integrated accuracy check.

2.6.9 Wire connection

All components have CAGE CLAMP® connections.

The WAGO CAGE CLAMP® connection is appropriate for solid, stranded and fine-stranded conductors. Each clamping unit accommodates one conductor.

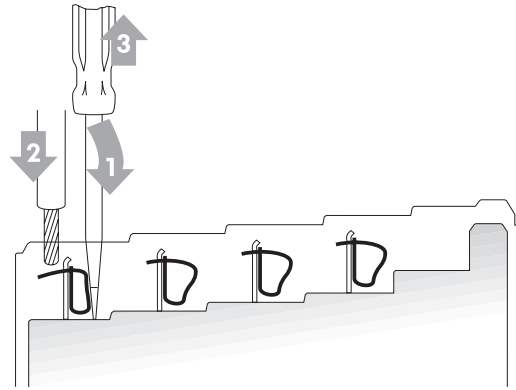


Fig. 2-9: CAGE CLAMP® Connection

g0xxx08x

The operating tool is inserted into the opening above the connection. This opens the CAGE CLAMP®. Subsequently the conductor can be inserted into the opening. After removing the operating tool, the conductor is safely clamped.

More than one conductor per connection is not permissible. If several conductors have to be made at one connection point, then they should be made away from the connection point using WAGO Terminal Blocks. The terminal blocks may be jumpered together and a single wire brought back to the I/O module connection point.



Attention

If it is unavoidable to jointly connect 2 conductors, then a ferrule must be used to join the wires together.

Ferrule:

Length	8 mm
Nominal cross section _{max.}	1 mm ² for 2 conductors with 0.5 mm ² each
WAGO Product	216-103 or products with comparable properties

2.7 Power Supply

2.7.1 Isolation

Within the fieldbus node, there are three electrically isolated potentials.

- Operational voltage for the fieldbus interface.
- Electronics of the couplers/controllers and the bus modules (internal bus).
- All bus modules have an electrical isolation between the electronics (internal bus, logic) and the field electronics. Some digital and analog input modules have each channel electrically isolated, please see catalog.

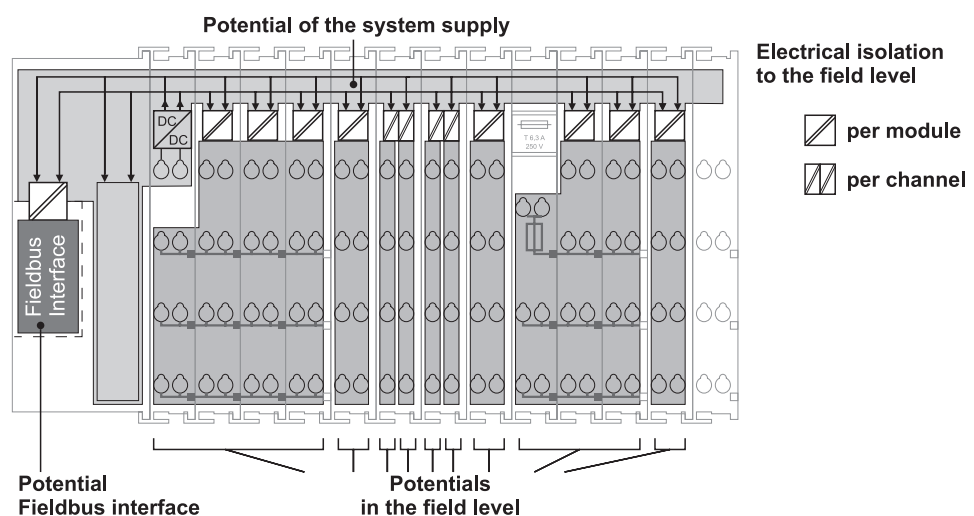


Fig. 2-10: Isolation

g0xx01e



Attention

The ground wire connection must be present in each group. In order that all protective conductor functions are maintained under all circumstances, it is recommended that a ground wire be connected at the beginning and end of a potential group. (ring format, please see chapter "2.8.3"). Thus, if a bus module comes loose from a composite during servicing, then the protective conductor connection is still guaranteed for all connected field devices.

When using a joint power supply unit for the 24 V system supply and the 24 V field supply, the electrical isolation between the internal bus and the field level is eliminated for the potential group.

2.7.2 System Supply

2.7.2.1 Connection

The WAGO-I/O-SYSTEM 750 requires a 24 V direct current system supply (-15% or +20%). The power supply is provided via the coupler/controller and, if necessary, in addition via the internal system supply modules (750-613). The voltage supply is reverse voltage protected.



Attention

The use of an incorrect supply voltage or frequency can cause severe damage to the component.

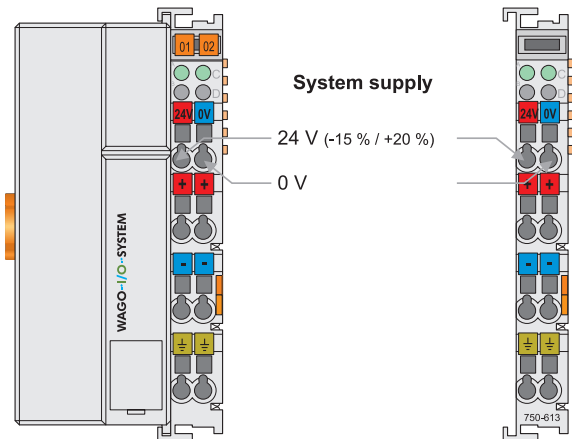


Fig. 2-11: System Supply

g0xxx02e

The direct current supplies all internal system components, e.g. coupler/controller electronics, fieldbus interface and bus modules via the internal bus (5 V system voltage). The 5 V system voltage is electrically connected to the 24 V system supply.

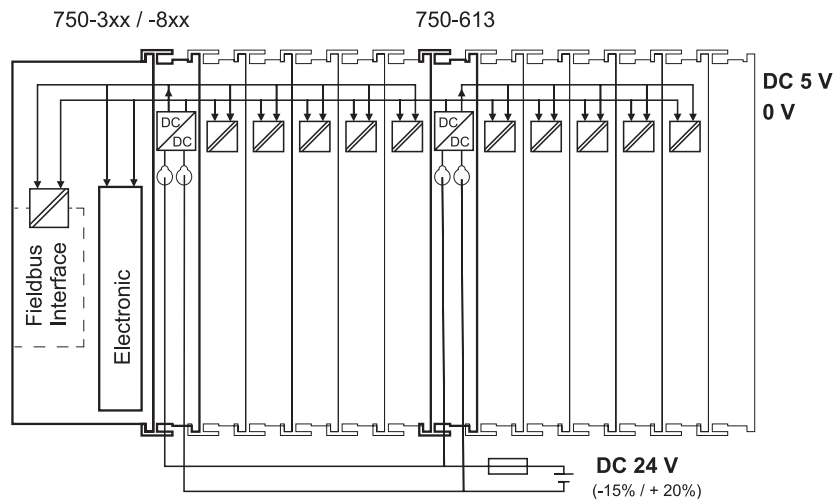


Fig. 2-12: System Voltage

g0xxx06e



Attention

Resetting the system by switching on and off the system supply, must take place simultaneously for all supply modules (coupler/controller and 750-613).

2.7.2.2 Alignment

Recommendation

A stable network supply cannot be taken for granted always and everywhere. Therefore, regulated power supply units should be used in order to guarantee the quality of the supply voltage.

The supply capacity of the coupler/controller or the internal system supply module (750-613) can be taken from the technical data of the components.

Internal current consumption*)	Current consumption via system voltage: 5 V for electronics of the bus modules and coupler/controller
Residual current for bus terminals*)	Available current for the bus modules. Provided by the bus power supply unit. See coupler/controller and internal system supply module (750-613)

*) cf. catalogue W4 Volume 3, manuals or Internet

Example

Coupler 750-301:
internal current consumption: 350 mA at 5V
residual current for
bus modules: 1650 mA at 5V
sum I(5V)_{total} : 2000 mA at 5V

The internal current consumption is indicated in the technical data for each bus terminal. In order to determine the overall requirement, add together the values of all bus modules in the node.



Attention

If the sum of the internal current consumption exceeds the residual current for bus modules, then an internal system supply module (750-613) must be placed before the module where the permissible residual current was exceeded.

Example:

A node with a PROFIBUS Coupler 750-333 consists of 20 relay modules (750-517) and 10 digital input modules (750-405).

Current consumption:
20* 90 mA = 1800 mA
10* 2 mA = 20 mA
Sum 1820 mA

The coupler can provide 1650 mA for the bus modules. Consequently, an internal system supply module (750-613), e.g. in the middle of the node, should be added.

Recommendation

With the WAGO ProServe® Software smartDESIGNER, the assembly of a fieldbus node can be configured. The configuration can be tested via the integrated accuracy check.

The maximum input current of the 24 V system supply is 500 mA. The exact electrical consumption ($I_{(24\text{ V})}$) can be determined with the following formulas:

Coupler/Controller

$$I(5\text{ V})_{\text{total}} = \text{Sum of all the internal current consumption of the connected bus modules} \\ + \text{internal current consumption coupler/controller}$$

750-613

$$I(5\text{ V})_{\text{total}} = \text{Sum of all the internal current consumption of the connected bus modules}$$

$$\text{Input current } I(24\text{ V}) = 5\text{ V} / 24\text{ V} * I(5\text{ V})_{\text{total}} / \eta \\ \eta = 0.87 \text{ (at nominal load)}$$



Note

If the electrical consumption of the power supply point for the 24 V-system supply exceeds 500 mA, then the cause may be an improperly aligned node or a defect.

During the test, all outputs, in particular those of the relay modules, must be active.

2.7.3 Field Supply

2.7.3.1 Connection

Sensors and actuators can be directly connected to the relevant channel of the bus module in 1-/4 conductor connection technology. The bus module supplies power to the sensors and actuators. The input and output drivers of some bus modules require the field side supply voltage.

The coupler/controller provides field side power (DC 24V). In this case it is a passive power supply without protection equipment.

Power supply modules are available for other potentials, e.g. AC 230 V. Likewise, with the aid of the power supply modules, various potentials can be set up. The connections are linked in pairs with a power contact.

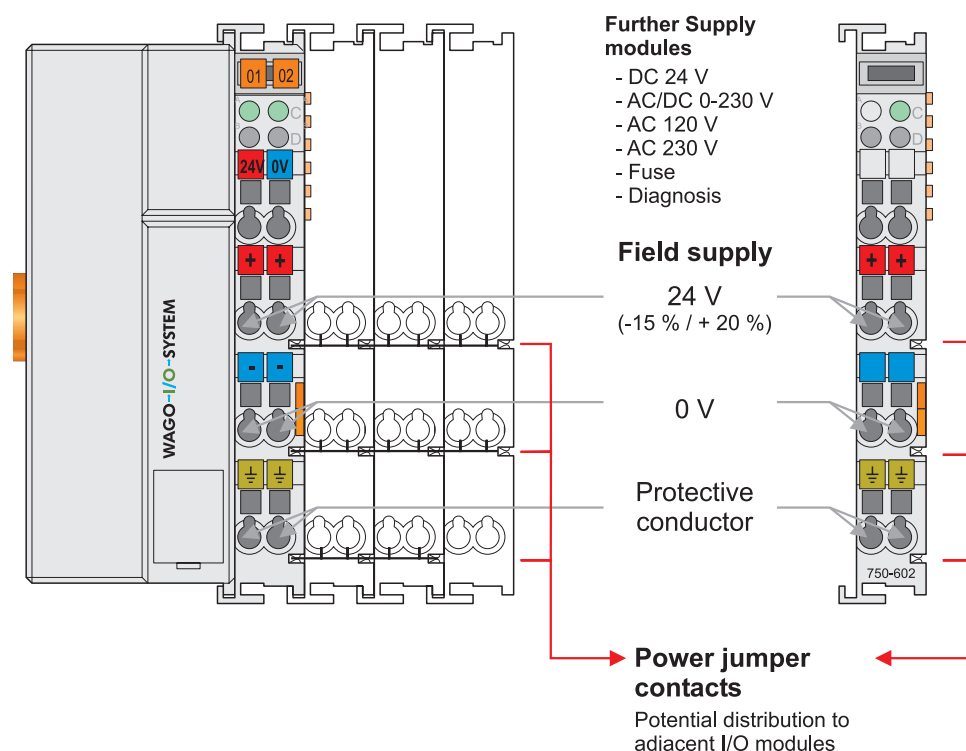


Fig. 2-13: Field Supply (Sensor/Actuator)

g0xxx03e

The supply voltage for the field side is automatically passed to the next module via the power jumper contacts when assembling the bus modules .

The current load of the power contacts must not exceed 10 A on a continual basis. The current load capacity between two connection terminals is identical to the load capacity of the connection wires.

By inserting an additional power supply module, the field supply via the power contacts is disrupted. From there a new power supply occurs which may also contain a new voltage potential.



Attention

Some bus modules have no or very few power contacts (depending on the I/O function). Due to this, the passing through of the relevant potential is disrupted. If a field supply is required for subsequent bus modules, then a power supply module must be used.

Note the data sheets of the bus modules.

In the case of a node setup with different potentials, e.g. the alteration from DC 24 V to AC 230V, a spacer module should be used. The optical separation of the potentials acts as a warning to heed caution in the case of wiring and maintenance works. Thus, the results of wiring errors can be prevented.

2.7.3.2 Fusing

Internal fusing of the field supply is possible for various field voltages via an appropriate power supply module.

750-601	24 V DC, Supply/Fuse
750-609	230 V AC, Supply/Fuse
750-615	120 V AC, Supply/Fuse
750-610	24 V DC, Supply/Fuse/Diagnosis
750-611	230 V AC, Supply/Fuse/Diagnosis

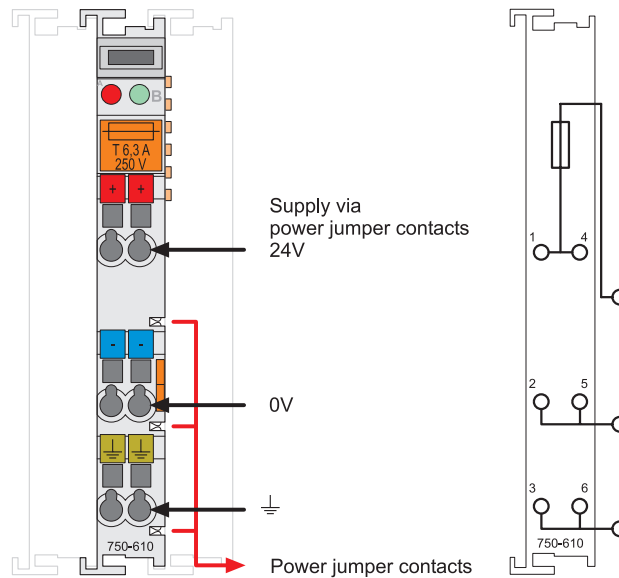


Fig. 2-14: Supply module with fuse carrier (Example 750-610)

g0xxx09x



Warning

In the case of power supply modules with fuse holders, only fuses with a maximum dissipation of 1.6 W (IEC 127) must be used.

For UL approved systems only use UL approved fuses.

In order to insert or change a fuse, or to switch off the voltage in succeeding bus modules, the fuse holder may be pulled out. In order to do this, use a screwdriver for example, to reach into one of the slits (one on both sides) and pull out the holder.



Fig. 2-15: Removing the fuse carrier

p0xxx05x

Lifting the cover to the side opens the fuse carrier.



Fig. 2-16: Opening the fuse carrier

p0xxx03x



Fig. 2-17: Change fuse

p0xxx04x

After changing the fuse, the fuse carrier is pushed back into its original position.

Alternatively, fusing can be done externally. The fuse modules of the WAGO series 281 and 282 are suitable for this purpose.

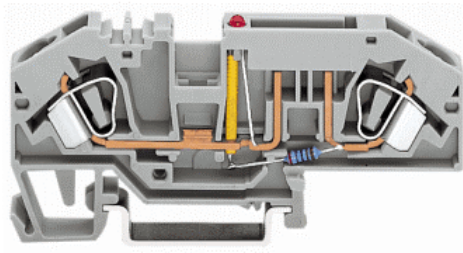


Fig. 2-18: Fuse modules for automotive fuses, Series 282

pf66800x

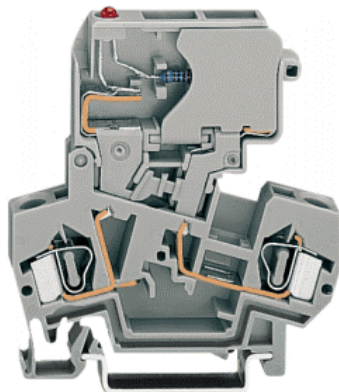


Fig. 2-19: Fuse modules with pivotable fuse carrier, Series 281

pe61100x

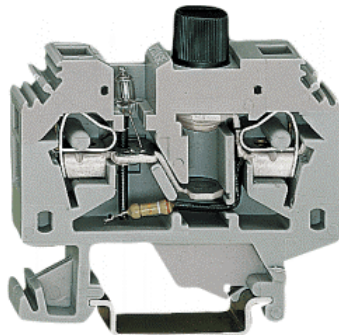


Fig. 2-20: Fuse modules, Series 282

pf12400x

2.7.4 Supplementary power supply regulations

The WAGO-I/O-SYSTEM 750 can also be used in shipbuilding or offshore and onshore areas of work (e.g. working platforms, loading plants). This is demonstrated by complying with the standards of influential classification companies such as Germanischer Lloyd and Lloyds Register.

Filter modules for 24-volt supply are required for the certified operation of the system.

Item No.	Name	Description
750-626	Supply filter	Filter module for system supply and field supply (24 V, 0 V), i.e. for field bus coupler/controller and bus power supply (750-613)
750-624	Supply filter	Filter module for the 24 V- field supply (750-602, 750-601, 750-610)

Therefore, the following power supply concept must be absolutely complied with.

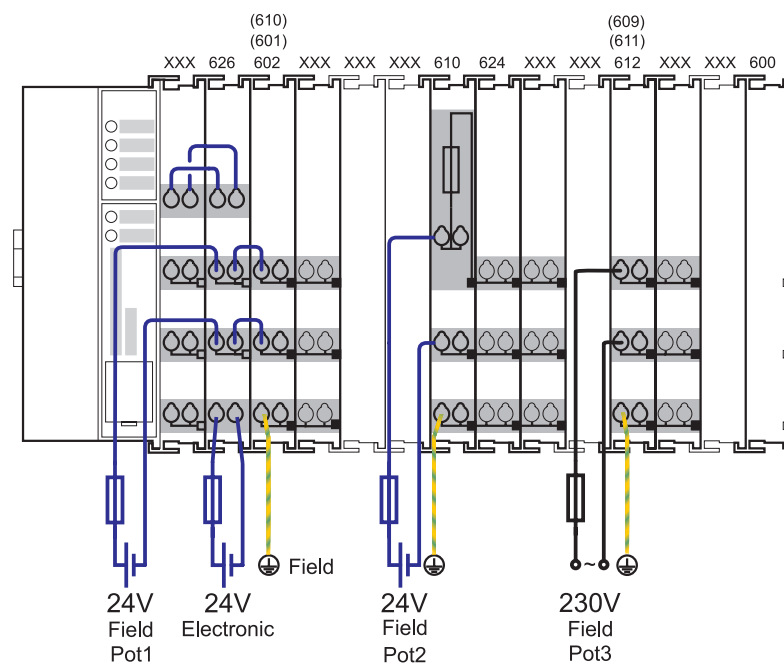


Fig. 2-21: Power supply concept

g01xx11e



Note

Another potential power terminal 750-601/602/610 must only be used behind the filter terminal 750-626 if the protective earth conductor is needed on the lower power contact or if a fuse protection is required.

2.7.5 Supply example



Note
The system supply and the field supply should be separated in order to ensure bus operation in the event of a short-circuit on the actuator side.

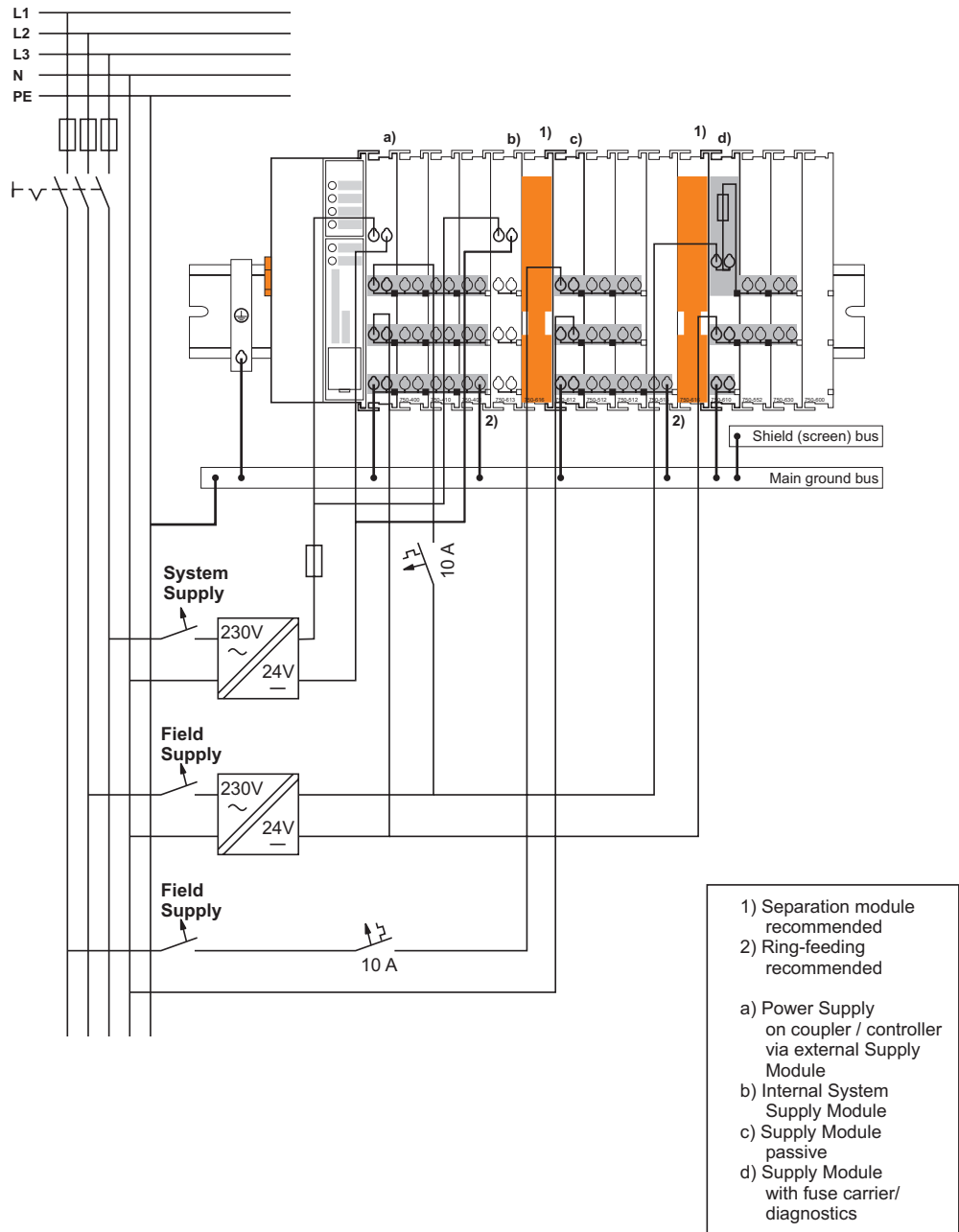


Fig. 2-22: Supply example

g0xxx04e

2.7.6 Power Supply Unit

The WAGO-I/O-SYSTEM 750 requires a 24 V direct current system supply with a maximum deviation of -15% or +20 %.

Recommendation

A stable network supply cannot be taken for granted always and everywhere. Therefore, regulated power supply units should be used in order to guarantee the quality of the supply voltage.

A buffer (200 µF per 1 A current load) should be provided for brief voltage dips. The I/O system buffers for approx 1 ms.

The electrical requirement for the field supply is to be determined individually for each power supply point. Thereby all loads through the field devices and bus modules should be considered. The field supply as well influences the bus modules, as the inputs and outputs of some bus modules require the voltage of the field supply.



Note

The system supply and the field supply should be isolated from the power supplies in order to ensure bus operation in the event of short circuits on the actuator side.

WAGO products Article No.	Description
787-903	Primary switched - mode, DC 24 V, 5 A wide input voltage range AC 85-264 V PFC (Power Factor Correction)
787-904	Primary switched - mode, DC 24 V, 10 A wide input voltage range AC 85-264 V PFC (Power Factor Correction)
787-912	Primary switched - mode, DC 24 V, 2 A wide input voltage range AC 85-264 V PFC (Power Factor Correction)
288-809 288-810 288-812 288-813	Rail-mounted modules with universal mounting carrier AC 115 V / DC 24 V; 0,5 A AC 230 V / DC 24 V; 0,5 A AC 230 V / DC 24 V; 2 A AC 115 V / DC 24 V; 2 A

2.8 Grounding

2.8.1 Grounding the DIN Rail

2.8.1.1 Framework Assembly

When setting up the framework, the carrier rail must be screwed together with the electrically conducting cabinet or housing frame. The framework or the housing must be grounded. The electronic connection is established via the screw. Thus, the carrier rail is grounded.



Attention

Care must be taken to ensure the flawless electrical connection between the carrier rail and the frame or housing in order to guarantee sufficient grounding.

2.8.1.2 Insulated Assembly

Insulated assembly has been achieved when there is constructively no direct conduction connection between the cabinet frame or machine parts and the carrier rail. Here the earth must be set up via an electrical conductor.

The connected grounding conductor should have a cross section of at least 4 mm².

Recommendation

The optimal insulated setup is a metallic assembly plate with grounding connection with an electrical conductive link with the carrier rail.

The separate grounding of the carrier rail can be easily set up with the aid of the WAGO ground wire terminals.

Article No.	Description
283-609	Single-conductor ground (earth) terminal block make an automatic contact to the carrier rail; conductor cross section: 0.2 -16 mm ² Note: Also order the end and intermediate plate (283-320)

2.8.2 Grounding Function

The grounding function increases the resistance against disturbances from electro-magnetic interferences. Some components in the I/O system have a carrier rail contact that dissipates electro-magnetic disturbances to the carrier rail.

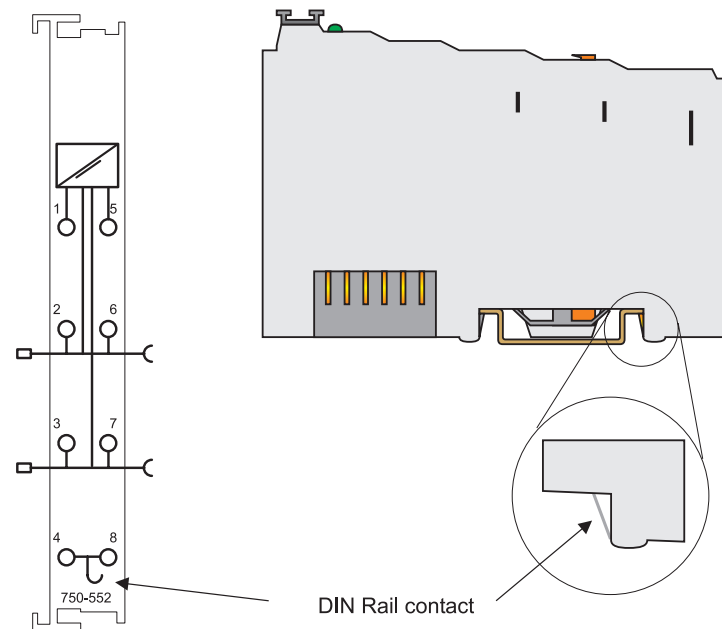


Fig. 2-23: Carrier rail contact

g0xxx10e



Attention

Care must be taken to ensure the direct electrical connection between the carrier rail contact and the carrier rail.

The carrier rail must be grounded.

For information on carrier rail properties, please see chapter 2.6.3.2.

2.8.3 Grounding Protection

For the field side, the ground wire is connected to the lowest connection terminals of the power supply module. The ground connection is then connected to the next module via the Power Jumper Contact (PJC). If the bus module has the lower power jumper contact, then the ground wire connection of the field devices can be directly connected to the lower connection terminals of the bus module.



Attention

Should the ground conductor connection of the power jumper contacts within the node become disrupted, e.g. due to a 4-channel bus terminal, the ground connection will need to be re-established.

The ring feeding of the grounding potential will increase the system safety. When one bus module is removed from the group, the grounding connection will remain intact.

The ring feeding method has the grounding conductor connected to the beginning and end of each potential group.

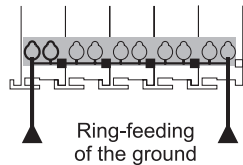


Fig. 2-24: Ring-feeding

g0xxx07e



Attention

The regulations relating to the place of assembly as well as the national regulations for maintenance and inspection of the grounding protection must be observed.

2.9 Shielding (Screening)

2.9.1 General

The shielding of the data and signal conductors reduces electromagnetic interferences thereby increasing the signal quality. Measurement errors, data transmission errors and even disturbances caused by overvoltage can be avoided.



Attention

Constant shielding is absolutely required in order to ensure the technical specifications in terms of the measurement accuracy.

The data and signal conductors should be separated from all high-voltage cables.

The cable shield should be potential. With this, incoming disturbances can be easily diverted.

The shielding should be placed over the entrance of the cabinet or housing in order to already repel disturbances at the entrance.

2.9.2 Bus Conductors

The shielding of the bus conductor is described in the relevant assembly guidelines and standards of the bus system.

2.9.3 Signal Conductors

Bus modules for most analog signals along with many of the interface bus modules include a connection for the shield.



Note

For better shield performance, the shield should have previously been placed over a large area. The WAGO shield connection system is suggested for such an application.

This suggestion is especially applicable when the equipment can have even current or high impulse formed currents running through it (for example through atmospheric end loading).

2.9.4 WAGO Shield (Screen) Connecting System

The WAGO Shield Connecting system includes a shield clamping saddle, a collection of rails and a variety of mounting feet. Together these allow many different possibilities. See catalog W4 volume 3 chapter 10.

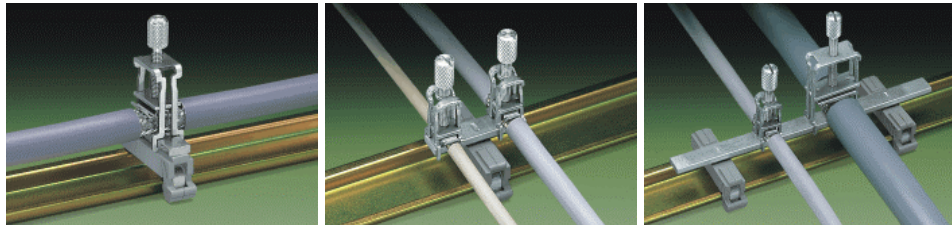


Fig. 2-25: WAGO Shield (Screen) Connecting System

p0xxx08x, p0xxx09x, and p0xxx10x

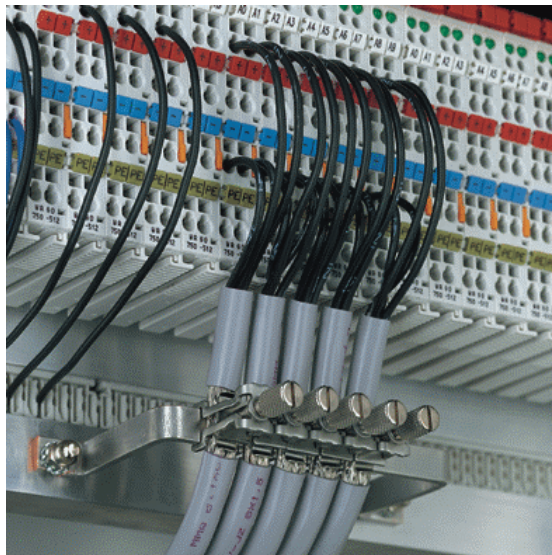


Fig. 2-26: Application of the WAGO Shield (Screen) Connecting System

p0xxx11x

2.10 Assembly Guidelines/Standards

DIN 60204,	Electrical equipping of machines
DIN EN 50178	Equipping of high-voltage systems with electronic components (replacement for VDE 0160)
EN 60439	Low voltage – switch box combinations

3 Check list (Draft)

Subject	Description	Chapter	Done
Project planning	1 coupler, ≤ 64 bus terminals, 1 end terminal (length of the nodes without cou- pler/controller ≤ 780 mm)	2.6.1	<input type="checkbox"/>
	sequence of the bus terminals mechanically feasible - power contacts	2.6.8	<input type="checkbox"/>
	sequence of the bus terminals electrically feasible - sum of electrical consumption $< 2A$	2.7.2	<input type="checkbox"/>
	...		<input type="checkbox"/>
Assembly	stable carrier rail	2.6.3	<input type="checkbox"/>
	end terminal for vertical assembly	2.6.1	<input type="checkbox"/>
	...		
Supply	separate power supply	2.7.1	<input type="checkbox"/>
	sum of electrical consumption $< 2 A$	2.7.2	<input type="checkbox"/>
	fusing of the field supply	2.7.3.2	<input type="checkbox"/>
	controlled power supply units	2.7.6	<input type="checkbox"/>
	...		
Grounding	carrier rail grounded	2.8.1	<input type="checkbox"/>
	ground terminal	2.8.1.2	<input type="checkbox"/>
	grounding function	2.8.2	<input type="checkbox"/>
	grounding protection	2.8.3	<input type="checkbox"/>
	ring feeding	2.8.3	<input type="checkbox"/>
	...		
Shielding	bus system shielded	2.9.2	<input type="checkbox"/>
	signal leads shielded	2.9.3	<input type="checkbox"/>
	...		



WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Telephone: 05 71/8 87 – 0
Fax: 05 71/8 87 – 1 69
E-Mail: info@wago.com

Internet: <http://www.wago.com>

This Page Left Intentionally Blank

Approvals Overview WAGO-I/O-SYSTEM 750/753

Approvals for versions with item no. suffix /...-... upon request.














	c UL us	E175199, UL 508
	ABS (American Bureau of Shipping)	03-HG374860/4-PDA; 10-HN17750340X
	BV (Bureau Veritas)	13453/CO BV, 30389/A0 BV
	DNV (Det Norske Veritas)	A-12306; Cl. B
	GL (Germanischer Lloyd)	11 631-10 HH; 19741-11 HH; 26 624-05 HH; 26 898-05 HH; 47 311-12 HH; 59 627-08 HH; 60 241-09 HH; 61 626-13 HH; Cat. A, B, C, D (EMC 1)
	KR (Korean Register of Shipping)	HMB05880-AC001, -AC002, -AC003, -AC004
	LR (Lloyd's Register)	02/20026 (E4); Env. 1, 2, 3, 4
	NKK (Nippon Kaiji Kyokai)	TA12716M
	Polski Rejestr Statkow	TE/1989/880590/13
	RINA (Registro Italiano Navale)	ELE134212XG
	c UL us	E198726, ANSI/ISA 12.12.01
	Brazilian Ex	TÜV 12.1297 X
	TÜV	07ATEX554086 X; IECEx TUN 09.0001 X 12ATEX106032 X; IECEx TUN 12.0039 X 14ATEX148929 X; IECEx TUN 14.0035 X



RINA PRS NKK LR KR GL DNV BV ABS

	Item No.	Item Description	Ex			Ship Approvals										UL									
	758-874	I/O-IPC-C6 Linux 2.6	x																			x			
	758-876	I/O-IPC-P14 Linux 2.6	x																				x		
	750-804	INTERBUS Controller	x	x	x																			x	
	750-806	DeviceNet Controller	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-815/300-000	MODBUS Controller, RS-485 (150 Bd ... 115,2 kbd)	x		x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-816/300-000	MODBUS Controller, RS-232 (150 Bd ... 115,2 kbd)	x		x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-830	BACnet/IP Controller	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-831	BACnet/IP Controller																							x
	750-833	PROFIBUS DP/V1 Controller	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-837	CANopen Controller, MSS	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-838	CANopen Controller, D-Sub	x		x	x	x	x	x		x		x		x		x		x		x		x		
	750-842	ETHERNET Controller 10 Mbit/s	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-843	ETHERNET Controller 10 Mbit/s	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-852	ETHERNET ECO Controller																							x
	750-871	ETHERNET TCP/IP Controller, 2 Ports	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-872	Telecontrol Controller RJ-45 + D-Sub	x		x																				x
	750-873	ETHERNET TCP/IP RS-232 Controller	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-880	ETHERNET Controller	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-881	ETHERNET Controller	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-882	Media Redundancy ETHERNET Controller	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-884	Application Controller BA	x																						x
	750-885	Media Redundancy ETHERNET Controller	x																						x
750-8202	PFC200 CS 2ETH RS	x																						x	
750-8203	PFC200 CS 2ETH CAN	x																						x	
750-8204	PFC200 CS 2ETH RS CAN	x																						x	
750-8206	PFC200 CS 2ETH RS CAN DPS	x																						x	

¹¹ Notice: The 750-626 filter module is absolutely necessary for ship approval (note the power supply instructions)!

	c UL us	E175199, UL 508
	ABS (American Bureau of Shipping)	03-HG374860/4-PDA; 10-HN17750340X
	BV (Bureau Veritas)	13453/CO BV, 30389/A0 BV
	DNV (Det Norske Veritas)	A-12306; Cl. B
	GL (Germanischer Lloyd)	11 631-10 HH; 19741-11 HH; 26 624-05 HH; 26 898-05 HH; 47 311-12 HH; 59 627-08 HH; 60 241-09 HH; 61 626-13 HH; Cat. A, B, C, D (EMC 1)
	KR (Korean Register of Shipping)	HMB05880-AC001, -AC002, -AC003, -AC004
	LR (Lloyd's Register)	02/20026 (E4); Env. 1, 2, 3, 4
	NKK (Nippon Kaiji Kyokai)	TA12716M
	Polski Rejestr Statkow	TE/1989/880590/13
	RINA (Registro Italiano Navale)	ELE134212XG
	c UL us	E198726, ANSI/ISA 12.12.01
	Brazilian Ex	TÜV 12.1297 X
	TÜV	07ATEX554086 X; IECEx TUN 09.0001 X 12ATEX106032 X; IECEx TUN 12.0039 X 14ATEX148929 X; IECEx TUN 14.0035 X

RINA PRS NKK LR KR GL DNV BV ABS

	Item No.	Item Description	Ex		Ship Approvals										UL			
	750-303	PROFIBUS DP-/FMS 12 Mbaud	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
	750-304	INTERBUS 500 kbaud	x	x													x	
	750-306	DeviceNet	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-307	CANopen	x	x													x	
	750-310	CC-Link	x	x													x	
	750-315/300-000	MODBUS, RS-485 (150 Bd ... 115,2 kbd)	x	x	x	x	x			x	x	x	x				x	
	750-316/300-000	MODBUS, RS-232 (150 Bd ... 115,2 kbd)	x	x	x	x	x			x	x	x	x				x	
	750-331	PROFIBUS DP 1.5 MBaud (Optical Fiber)																x
	750-333	PROFIBUS DP/V1 12 MBaud	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-334	INTERBUS 500 kBaud (Optical Fiber)		x	x													x
	750-337	CANopen, MSS	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-338	CANopen, D-Sub	x	x	x	x	x				x							x
	750-340	PROFINET IO Coupler	x															x
	750-342	ETHERNET TCP/IP 10 Mbit/s	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-343	PROFIBUS DP ECO 12 Mbaud	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-344	INTERBUS ECO 500 kbaud	x	x	x													x
	750-345	INTERBUS ECO 2 Mbaud	x	x	x													x
	750-346	DeviceNet ECO	x	x	x													x
	750-347	CANopen ECO, MSS	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-348	CANopen ECO	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-351	SERCOS III Coupler	x		x	x	x			x	x	x	x	x	x	x	x	
	750-352	ETHERNET TCP/IP 10/100 Mbit/s	x		x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-354	EtherCAT®	x		x													x
750-370	PROFINET IO 100Mbit 2-port	x		x													x	
750-375	PROFINET IO adv. 2-Port																x	
750-377	PROFINET IO adv. ECO 2-Port																x	
	750-960	PROFIBUS Fieldbus connector with D-Sub male connector	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-961	INTERBUS Fieldbus connector with D-Sub female connector	x		x													x
	750-962	INTERBUS Fieldbus connector with D-Sub male connector	x		x													x
	750-963	CANopen Fieldbus connector with D-Sub connectors	x			x	x	x	x	x	x	x	x	x	x	x	x	
	750-970	PROFIBUS Fieldbus connector with D-Sub female connector			x	x	x	x	x	x	x	x	x	x	x	x	x	
	750-975	ETHERNET RJ-45 connector				x	x	x	x	x	x	x	x	x	x	x		
	750-976	PROFINET RJ-45 connector																x

¹⁾ Notice: The 750-626 filter module is absolutely necessary for ship approval (note the power supply instructions)!



c UL us

E175199, UL 508



ABS (American Bureau of Shipping)

03-HG374860/4-PDA; 10-HN17750340X



BV (Bureau Veritas)

13453/CO BV, 30389/AO BV



DNV (Det Norske Veritas)

A-12306; Cl. B



GL (Germanischer Lloyd)

11 631-10 HH; 19741-11 HH; 26 624-05 HH; 26 898-05 HH; 47 311-12 HH; 59 627-08 HH; 60 241-09 HH; 61 626-13 HH; Cat. A, B, C, D (EMC 1)



KR (Korean Register of Shipping)

HMB05880-AC001, -AC002, -AC003, -AC004



LR (Lloyd's Register)

02/20026 (E4); Env. 1, 2, 3, 4



NKK (Nippon Kaiji Kyokai)

TA12716M



Polski Rejestr Statkow

TE/1989/880590/13



RINA (Registro Italiano Navale)

ELE134212XG

c UL us

E198726, ANSI/ISA 12.12.01




Brazilian Ex

TÜV 12.1297 X

TÜV

07ATEX554086 X; IECEx TUN 09.0001 X
12ATEX106032 X; IECEx TUN 12.0039 X
14ATEX148929 X; IECEx TUN 14.0035 X

RINA PRS NKK LR KR GL DNV BV ABS

	Item No.	Item Description	Ex		Ship Approvals										UL		
					RINA	PRS	NKK	LR	KR	GL	DNV	BV	ABS				
Digital Input Modules 	750-400	2DI 24V DC, 3.0ms	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-401	2DI 24V DC, 0.2ms	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-402	4DI 24V DC, 3.0ms	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-403	4DI 24V DC, 0.2ms	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-405	2DI 230V AC	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-406	2DI 120V AC	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-407	2DI 220 V DC															x
	750-408	4DI 24V DC, 3.0ms, low-side switching	x*	x*	x*	x*	x*	x*			x*				x*		x*
	750-409	4DI 24V DC, 0.2ms, low-side switching	x*	x*													x*
	750-410	2DI 24V DC, 3.0ms, proximity switching	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-411	2DI 24V DC, 0.2ms, proximity switching	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-412	2DI 48V DC, 3.0ms	x*	x*													x*
	750-414	4DI 5V DC, 0.2ms		x													x
	750-415	4DI 24V AC/DC 20ms	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-418	2DI 24V DC, 3.0ms, diagn., acknol.	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-421	2DI 24V DC, diagn.	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-422	2DI 24V DC, pulse extension	x*	x	x*												x*
	750-423	4DI 24V AC/DC, 50ms, power contacts	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-424	2DI 24V DC, intruder detection	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-425	2DI NAMUR	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-427	2DI 110V DC	x*	x	x*												x*
	750-428	4DI 42V AC/DC, 20ms	x*	x	x*												x*
	750-430	8DI 24V DC, 3.0ms	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-431	8DI 24V DC, 0.2ms	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-432	4DI 24V DC	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-433	4DI 24V DC	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-435 ²⁾	1DI NAMUR Exi	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-436	8DI 24V DC, 3.0ms	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-437	8DI 24V DC, 0.2ms	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-438 ²⁾	2DI NAMUR Exi	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-439 ²⁾	8DI NAMUR Exi	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1400	16DI 24V DC 3.0ms, high-side switch., ribbon cable	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1402	16DI 24V DC 3.0ms, low-side switch., ribbon cable	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1405	16DI 24V DC 3.0ms, high-side switch.	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1406	16DI 24V DC 0.2ms, high-side switch.	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1407	16DI 24V DC 3.0ms, low-side switch.	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1415	8DI 24V DC, 3.0ms, high-side switch., 2-conductor	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1416	8DI 24V DC, 0.2ms, high-side switch., 2-conductor	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1417	8DI 24V DC, 3.0ms, low-side switch., 2-conductor	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1418	8DI 24V DC, 0.2ms, low-side switch., 2-conductor	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1420	4DI 24V DC, 3.0ms, high-side switch., 3-conductor	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1421	4DI 24V DC, 0.2ms, high-side switch., 3-conductor	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1422	4DI 24V DC, 3.0ms, low-side switch., 3-conductor	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
750-1423	4DI 24V DC, 0.2ms, low-side switch., 3-conductor	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
750-1425	8DI PTC																
753-429	2DI 60 V DC, 3.0 ms	x	x													x	
753-434	8DI 5/12V DC	x	x													x	
753-440	4DI 120/230V AC	x	x													x	

* Approval is also granted for the 753 Series pluggable I/O modules

²⁾ Only use this module in connection with the 24VDC Ex i supply module (note the power supply instructions!)














	c UL us	E175199, UL 508
	ABS (American Bureau of Shipping)	03-HG374860/4-PDA; 10-HN17750340X
	BV (Bureau Veritas)	13453/CO BV, 30389/AO BV
	DNV (Det Norske Veritas)	A-12306; Cl. B
	GL (Germanischer Lloyd)	11 631-10 HH; 19741-11 HH; 26 624-05 HH; 26 898-05 HH; 47 311-12 HH; 59 627-08 HH; 60 241-09 HH; 61 626-13 HH; Cat. A, B, C, D (EMC 1)
	KR (Korean Register of Shipping)	HMB05880-AC001, -AC002, -AC003, -AC004
	LR (Lloyd's Register)	02/20026 (E4); Env. 1, 2, 3, 4
	NKK (Nippon Kaiji Kyokai)	TA12716M
	Polski Rejestr Statkow	TE/1989/880590/13
	RINA (Registro Italiano Navale)	ELE134212XG
	c UL us	E198726, ANSI/ISA 12.12.01
	Brazilian Ex	TÜV 12.1297 X
	TÜV	07ATEX554086 X; IECEx TUN 09.0001 X 12ATEX106032 X; IECEx TUN 12.0039 X 14ATEX148929 X; IECEx TUN 14.0035 X

RINA PRS NKK LR KR GL DNV BV ABS

	Item No.	Item Description	Ex		Ship Approvals										UL											
					RINA	PRS	NKK	LR	KR	GL	DNV	BV	ABS													
Digital Output Modules	750-501	2DO 24V DC, 0.5A	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	
	750-502	2DO 24V DC, 2.0A	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-504	4DO 24V DC, 0.5A	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-506	2DO 24V DC, 0.5A, diagn.	x*	x*																						x*
	750-508	2DO 24V DC, 2.0A, diagn.	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-509	2DO 230V AC, 0.3A, SSR	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-512	2DO 230V AC, 2.0A, rel. 2 NO	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-513	2DO 230V AC, 2.0A, rel. 2 NO, pot. free	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-514	2DO 125V AC, 0.5A, rel. 2 CO, pot. free	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-516	4DO 24V DC, 0.5A, low side switch	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-517	2DO 230V AC, 1.0A, rel. 2 CO, pot. free	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-519	4DO 5V DC, 20mA	x	x	x																					x
	750-522	2DO 230V AC, 3.0A, 30s, SSR	x	x	x																					x
	750-523	1DO 230V AC, 16A, rel. 1a, pot. free				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-530	8DO 24V DC, 0.5A	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-531	4DO 24V DC, 0.5A	x*		x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-532	4DO 24V DC, 0.5A, diagn.	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-534	8DO 5/12V DC	x*		x*																					x*
	750-535 ²⁾	2DO 24V DC, Ex i	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-536	8DO 24V DC, 0.5A	x*		x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-537	8DO 24V DC, 0.5A, diagn.	x*		x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-538 ²⁾	2DO RELAY Ex i	x		x																					x
	750-1500	16DO 24V DC 0.5A, high-side switch., ribbon cable	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1501	16DO 24V DC 0.5A, low-side switch., ribbon cable	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1502	8DI 8DO 24V DC 0.5A, high-side switch., ribbon cable	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1504	16DO 24V DC 0.5A, high-side switch.	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
750-1505	16DO 24V DC 0.5A, low-side switch.	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
750-1506	8DI 8DO 24V DC 0.5A, high-side switch.	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
750-1515	8DO 24V DC 0.5A, high-side switch., 2-conductor	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
750-1516	8DO 24V DC 0.5A, low-side switch., 2-conductor	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
753-540	4DO 120/230V AC	x		x																					x	















* Approval is also granted for the 753 Series pluggable I/O modules
²⁾ Only use this module in connection with the 24VDC Ex i supply module (note the power supply instructions!)

	c UL us	E175199, UL 508
	ABS (American Bureau of Shipping)	03-HG374860/4-PDA; 10-HN17750340X
	BV (Bureau Veritas)	13453/CO BV, 30389/AO BV
	DNV (Det Norske Veritas)	A-12306; Cl. B
	GL (Germanischer Lloyd)	11 631-10 HH; 19741-11 HH; 26 624-05 HH; 26 898-05 HH; 47 311-12 HH; 59 627-08 HH; 60 241-09 HH; 61 626-13 HH; Cat. A, B, C, D (EMC 1)
	KR (Korean Register of Shipping)	HMB05880-AC001, -AC002, -AC003, -AC004
	LR (Lloyd's Register)	02/20026 (E4); Env. 1, 2, 3, 4
	NKK (Nippon Kaiji Kyokai)	TA12716M
	Polski Rejestr Statkow	TE/1989/880590/13
	RINA (Registro Italiano Navale)	ELE134212XG
	c UL us	E198726, ANSI/ISA 12.12.01
	Brazilian Ex	TÜV 12.1297 X
	TÜV	07ATEX554086 X; IECEx TUN 09.0001 X 12ATEX106032 X; IECEx TUN 12.0039 X 14ATEX148929 X; IECEx TUN 14.0035 X
		RINA PRS NKK LR KR GL DNV BV ABS

Analog Input Modules	Item No.	Item Description	Ex		Ship Approvals										UL	
					RINA	PRS	NKK	LR	KR	GL	DNV	BV	ABS			
	750-452	2AI 0 - 20mA, differential input	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-453	4AI 0 - 20mA, single ended	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-454	2AI 4 - 20mA, differential input	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-455	4AI 4 - 20mA, single ended	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-456	2AI ± 10 VDC	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-457	4AI ± 10 VDC, single ended	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-459	4AI 0-10 VDC, single ended	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-460	4AI Resistance Temperature Device (RTD)														x
	750-461	2AI Resistance Temperature Device (RTD)	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-463	4 AI RTD/-30°C ... +150°C	x													x
	750-464	2/4AI RTD, configurable	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-465	2AI 0 - 20mA, single ended	x*	x*												x*
	750-466	2AI 4 - 20mA, single ended	x*	x*												x*
	750-467	2AI 0 - 10 VDC, single ended	x*	x*												x*
	750-468	4AI 0 - 10 VDC, single ended	x	x	x	x			x	x	x	x	x	x	x	x
	750-469	2AI Thermocouple (TC) diagn.	x*	x*	x*	x*			x*	x*	x*	x*	x*	x*	x*	x*
	750-470	2AI 0 - 20mA, single ended, s.c. protec.	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-472	2AI 0 - 20mA, 16 bits, single ended	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-473	2AI 4 - 20mA, single ended, s.c. protec.	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-474	2AI 4 - 20mA, 16 bits, single ended	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-475	2AI 0 - 1A AC/DC, differential input	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-476	2AI ± 10 VDC, 16 bits, single ended	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-477	2AI 0 - 10V AC/DC, differential input	x*	x	x*											x*
	750-478	2AI 0 - 10 VDC, 16 bits, single ended	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-479	2AI ± 10 VDC, differential measurement input	x*	x	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-480	2AI 0 - 20mA, differential measurement input	x*	x	x*											x*
	750-481/003-000 ²⁾	2AI RTD, Ex i	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-482	2AI 4 - 20 mA 12 Bit single ended HART	x*	x*												x*
	750-483	2AI 0 - 30 VDC, differential measurement input	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-484 ²⁾	2AI 4 - 20 mA 12 Bit single ended HART Ex i	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-485	2AI 4 - 20mA, Ex i	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-487/003-000 ²⁾	2AI TC, Ex i	x	x	x	x	x	x								x
	750-492	2AI 4 - 20mA, differential measurement input	x*	x	x*											x*
	750-493	3-Phase Power Measurement Module	x													x
	750-494	3-Phase Power Measurement Module	x													x

* Approval is also granted for the 753 Series pluggable I/O modules

²⁾ Only use this module in connection with the 24VDC Ex i supply module (note the power supply instructions!)

	c UL us	E175199, UL 508
	ABS (American Bureau of Shipping)	03-HG374860/4-PDA; 10-HN17750340X
	BV (Bureau Veritas)	13453/CO BV, 30389/AO BV
	DNV (Det Norske Veritas)	A-12306; Cl. B
	GL (Germanischer Lloyd)	11 631-10 HH; 19741-11 HH; 26 624-05 HH; 26 898-05 HH; 47 311-12 HH; 59 627-08 HH; 60 241-09 HH; 61 626-13 HH; Cat. A, B, C, D (EMC 1)
	KR (Korean Register of Shipping)	HMB05880-AC001, -AC002, -AC003, -AC004
	LR (Lloyd's Register)	02/20026 (E4); Env. 1, 2, 3, 4
	NKK (Nippon Kaiji Kyokai)	TA12716M
	Polski Rejestr Statkow	TE/1989/880590/13
	RINA (Registro Italiano Navale)	ELE134212XG
	c UL us	E198726, ANSI/ISA 12.12.01
	Brazilian Ex	TÜV 12.1297 X
	TÜV	07ATEX554086 X; IECEx TUN 09.0001 X 12ATEX106032 X; IECEx TUN 12.0039 X 14ATEX148929 X; IECEx TUN 14.0035 X












RINA PRS NKK LR KR GL DNV BV ABS

	Item No.	Item Description	Ex		Ship Approvals										UL				
Analog Output Modules	750-550	2AO 0 - 10 VDC	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*		
	750-552	2AO 0 - 20mA	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	
	750-553	4AO 0 - 20mA	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	
	750-554	2AO 4 - 20mA	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	
	750-555	4AO 4 - 20mA	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	
	750-556	2AO ±10 VDC	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	
	750-557	4AO ±10 VDC	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	
	750-559	4AO 0-10 VDC	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	
	750-560	2 AO 0-10 VDC 10 bits 10 mA 24V	x	x														x	
	750-562	2AO 0/±10 VDC 16 bits	x	x															x
	750-563	2AO 0/4-20 mA / 6-18 VDC 16 bits	x	x															x
	750-585 ²⁾	2AO 0- 20mA, Ex i	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-586 ²⁾	2AO 4- 20mA, Ex i	x	x															x



* Approval is also granted for the 753 Series pluggable I/O modules

²⁾ Only use this module in connection with the 24VDC Ex i supply module (note the power supply instructions!)

	c UL us	E175199, UL 508
	ABS (American Bureau of Shipping)	03-HG374860/4-PDA; 10-HN17750340X
	BV (Bureau Veritas)	13453/CO BV, 30389/AO BV
	DNV (Det Norske Veritas)	A-12306; Cl. B
	GL (Germanischer Lloyd)	11 631-10 HH; 19741-11 HH; 26 624-05 HH; 26 898-05 HH; 47 311-12 HH; 59 627-08 HH; 60 241-09 HH; 61 626-13 HH; Cat. A, B, C, D (EMC 1)
	KR (Korean Register of Shipping)	HMB05880-AC001, -AC002, -AC003, -AC004
	LR (Lloyd's Register)	02/20026 (E4); Env. 1, 2, 3, 4
	NKK (Nippon Kaiji Kyokai)	TA12716M
	Polski Rejestr Statkow	TE/1989/880590/13
	RINA (Registro Italiano Navale)	ELE134212XG
c UL us		E198726, ANSI/ISA 12.12.01
	Brazilian Ex	TÜV 12.1297 X
TÜV		07ATEX554086 X; IECEx TUN 09.0001 X 12ATEX106032 X; IECEx TUN 12.0039 X 14ATEX148929 X; IECEx TUN 14.0035 X

RINA PRS NKK LR KR GL DNV BV ABS

Function, Technology and Communication Modules	Item No.	Item Description	Ex		Ship Approvals										UL	
					RINA	PRS	NKK	LR	KR	GL	DNV	BV	ABS			
	750-404	Up/down counter, 100kHz	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-511	2 DO, 24 VDC, 0.1A, pulse width	x	x												x
	750-630	SSI transmitter interface	x	x												x
	750-631/000-004	Incremental encoder interface, 16 bits		x												x
	750-632	Proportional Valve Module	x													x
	750-633 2)	Up/down counter Ex i	x	x												x
	750-635	Digital pulse interface	x*	x	x*											x*
	750-637	Incremental encoder interface, 32 bits	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-638	Up/down counter, 500kHz	x*	x	x*											x*
	750-640	RTC Module	x	x												x
	750-642	Radio receiver module	x	x												x
	750-643	MP-Bus Master module	x	x												x
	750-644	Bluetooth®/RF Transceiver	x	x												x
	750-645	2AI/2DO VIB VRMS/SPM Multi	x	x												x
	750-650	Serial interface RS-232 C	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-651	TTY interface		x	x	x	x	x	x	x	x	x	x	x	x	x
	750-652	RS-232/RS-485 configurable	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-653	Serial interface RS-485	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-654	Data exchange module			x	x	x	x	x	x	x	x	x	x	x	x
	750-655	AS-Interface Master	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-657	4-Channel IO-Link Master														x
	750-658	CAN Gateway														x
	750-660/000-001	PROFIsafe 8 DI	x	x												x
	750-661/000-003	PROFIsafe 4FDI 24V	x*	x*												x*
	750-662/000-003	PROFIsafe 8 FDI 24 V	x*	x*												x*
	750-663/000-003	4F Ex i DI 24V PROFIsafe V2 iPar	x	x												x
	750-665/000-001	PROFIsafe 4 FDO 0.5 A; 4 FDI 24V DC	x	x												x
	750-666/000-003	PROFIsafe 4 FDI/2 FDO 24V/10A	x*													x*
	750-667/000-003	PROFIsafe 4 FDI/4 FDO 24V/2A	x*	x*												x*
	750-670	Stepper Controller RS-422/24V/20mA	x	x												x
	750-671	Stepper Controller 24V/1.5A	x													x
	753-646	KNX/EIB/TP1 module				x	x	x	x	x	x	x	x	x	x	x
	753-647	DALI Multi-Master Module														x
	753-648	LON FTT Module														x



* Approval is also granted for the 753 Series pluggable I/O modules
 2) Only use this module in connection with the 24VDC Ex i supply module (note the power supply instructions!)

	c UL us	E175199, UL 508
	ABS (American Bureau of Shipping)	03-HG374860/4-PDA; 10-HN17750340X
	BV (Bureau Veritas)	13453/CO BV, 30389/AO BV
	DNV (Det Norske Veritas)	A-12306; Cl. B
	GL (Germanischer Lloyd)	11 631-10 HH; 19741-11 HH; 26 624-05 HH; 26 898-05 HH; 47 311-12 HH; 59 627-08 HH; 60 241-09 HH; 61 626-13 HH; Cat. A, B, C, D (EMC 1)
	KR (Korean Register of Shipping)	HMB05880-AC001, -AC002, -AC003, -AC004
	LR (Lloyd's Register)	02/20026 (E4); Env. 1, 2, 3, 4
	NKK (Nippon Kaiji Kyokai)	TA12716M
	Polski Rejestr Statkow	TE/1989/880590/13
	RINA (Registro Italiano Navale)	ELE134212XG
	c UL us	E198726, ANSI/ISA 12.12.01
	Brazilian Ex	TÜV 12.1297 X
	TÜV	07ATEX554086 X; IECEx TUN 09.0001 X 12ATEX106032 X; IECEx TUN 12.0039 X 14ATEX148929 X; IECEx TUN 14.0035 X

RINA PRS NKK LR KR GL DNV BV ABS

Supply and Segment Modules	Item No.	Item Description	Ex		Ship Approvals										UL			
	750-600	End module	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-601 ³⁾	Supply module 24 VDC /fuse	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-602 ³⁾	Supply module 24 VDC	x*		x*	x*	x*	x*	x*	x	x*	x*	x*	x*	x	x*	x*	x*
	750-603	Field side connection module	x*		x*	x*	x*		x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-604	Field side connection module	x*		x*	x*	x*		x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-606	Supply module 24 VDC Ex i	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-609	Supply module 230 VAC /fuse	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-610 ³⁾	Supply module 24 VDC /fuse /diagn.	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-611	Supply module 230 VDC /fuse /diagn.	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-612 ⁴⁾	Supply module 0-230 V AC/DC	x*		x*	x*	x*	x*	x	x*	x*	x*	x*	x	x*	x*	x*	x*
	750-613 ¹⁾	Internal system supply module 24 VDC	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-614	Field side connection module, 0 - 230 V AC/DC	x*		x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*
	750-615	Supply module 120 VAC /fuse	x		x													x
	750-616	Separation module	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-617	Supply module 24 VAC /fuse			x													x
	750-621	Separation module with contacts	x		x													x
	750-622	Binary spacer module with power supply	x		x													x
	750-623	Supply module 24/5-15 VDC	x		x	x	x	x	x		x		x					x
	750-624	Overtoltage protection, field side power supply	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-625/000-001	Supply module 24 VDC Ex i	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-626	Filter module, system and field side power supply	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-627	Internal Data Bus Extension End Module												x				x
	750-628	Internal Data Bus Extension Coupler Module												x				x
	750-1605	Field side connection module	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1606	Field side connection module	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	750-1607	Field side connection module	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
	753-620	DALI Multi-Master DC/DC Converter																x
	753-629/020-000	Spacer Module passive																x
	753-1629	Spacer Module active																x

* Approval is also granted for the 753 Series pluggable I/O modules

¹⁾ Notice: The 750-626 filter module is absolutely necessary for ship approval (note the power supply instructions)!

³⁾ Notice: The 750-624 filter module is absolutely necessary for ship approval (note the power supply instructions)!

⁴⁾ Notice: The 750-626 filter module is absolutely necessary for ship approval at 24VDC power supply (note the power supply instructions)!

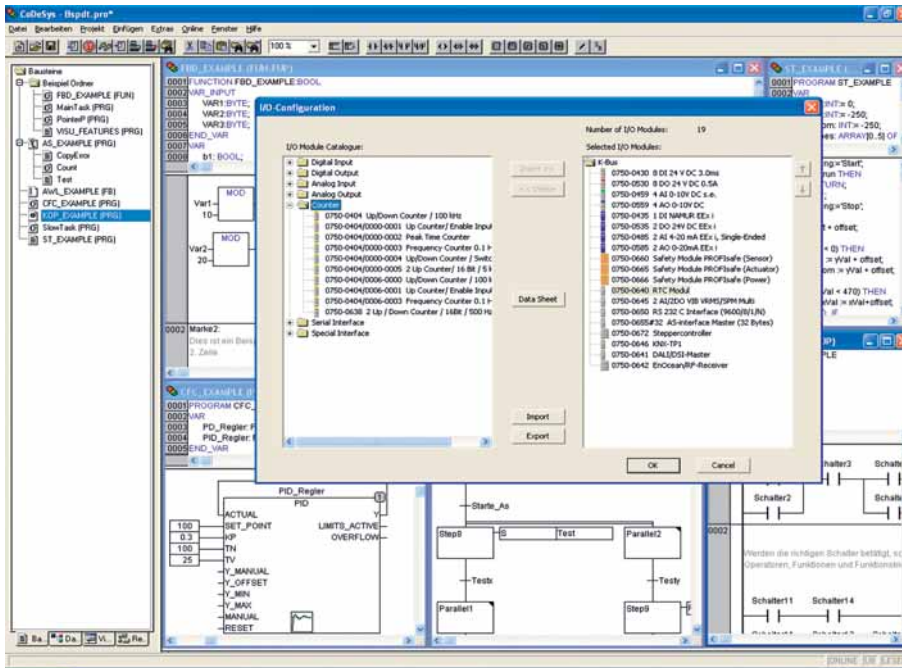
This Page Left Intentionally Blank

WAGO SYSTEM 750 PRO- **SOFTWARE**

Operations & Maintenance Manual
December 2015

WAGO-I/O-PRO V2.3

IEC 61131-3 programming tool



WAGO-I/O-PRO is a programming and visualization tool for control programs. This software is used to develop PLC applications for WAGO-I/O-SYSTEM 750 Programmable Fieldbus Controllers.

WAGO-I/O-PRO runs in line with the IEC 61131-3 standard. This standard describes the requirements of a programming system. The IL, SFC, LD, FBD and ST programming languages are supported. The optimal programming language can be chosen for each application.

With extensive programming functions, the software readily meets the increasing requirements of control program development (e.g., reusability and modularization).

- Highly efficient translation between programming languages
- Automatic declaration of variables
- Library management

Integrated test and diagnostic functions also streamline and accelerate the implementation of processes for PLC projects.

- Online status indication in program code
- Offline simulation
- Integrated process visualization
- Recording and graphical presentation of project variables

WAGO-I/O-PRO also offers the option of programming products from other manufacturers within the CODESYS automation alliance in addition to the standard programmable CODESYS automation alliance products from WAGO.

Description	Item No.	Pack. Unit	Technical Data
WAGO-I/O-PRO V2.3, RS-232 Kit	759-333	1	Supported operating systems Windows XP (SP3 or later), Windows 7 System requirements Processor 1 GHz or higher 32-bit (x86) or 64-bit (x64) Memory min. 1 GB RAM 2 GB RAM or greater recommended Hard disk storage min. 300 MB CD-ROM required Graphics resolution min. 1024 x 786 1280 x 1024 or higher recommended Mouse required Miscellaneous Open serial interface Delivery type RS-232 Kit: CD-ROM with software and serial 750-920 Communication Cable USB Kit: CD-ROM with software and USB communication cable 750-923 CD-ROM: CD-ROM with software, no communication cable
WAGO-I/O-PRO V2.3, USB Kit	759-333/000-923	1	
WAGO-I/O-PRO V2.3, CD-ROM	759-911	1	
Approvals			
Shipbuilding	ABS, DNV, GL, KR, NKK, RINA		

This Page Left Intentionally Blank



WAGO-I/O-SYSTEM 750 **IEC 61850 Solution for programmable Controls** **of Telecontrol Technology** **759-911**

Version 1.2.2



© 2015 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

1	Notes about this Documentation.....	5
1.1	Copyright.....	5
1.2	Symbols.....	6
1.3	Number Notation.....	8
1.4	Font Conventions.....	8
2	Important Notes	9
2.1	Legal Bases	9
2.1.1	Subject to Changes	9
2.1.2	Personnel Qualifications.....	9
2.1.3	Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions	9
2.1.4	Technical Condition of Specified Devices.....	10
2.2	Safety Advice (Precautions).....	11
3	Application Description.....	13
3.1	Abbreviations	13
3.2	Tasks	14
3.3	Hardware Requirements.....	15
3.4	Software Requirements	15
3.5	Requirements and Capabilities.....	15
3.6	MMS Communication.....	16
3.6.1	Create New Project and Open the IEC 61850 IEC 61850 Configurator.....	16
3.6.2	Create an IEC 61850 Configuration.....	17
3.6.3	Create Dataset.....	20
3.6.4	Create a Report.....	21
3.6.5	Generate Code, Translate and Load Solution into the Controller	22
3.6.6	Connect WAGO Controller with IEC 61850 Client.....	23
3.6.7	Conclusion.....	25
3.7	GOOSE Communication.....	26
3.7.1	Configure Publisher.....	26
3.7.2	Configure Subscriber.....	34
3.7.3	Commission the Configuration.....	35
4	Appendix.....	38
4.1	IEC 61850 Configurator.....	38
4.1.1	Overview	38
4.1.2	Menu Bar	39
4.1.2.1	“File” Menu	39
4.1.2.2	“Tools” Menu	40
4.1.2.3	“Help” Menu.....	41
4.1.3	Operating range	42
4.1.3.1	“Configuration” Tab	42
4.1.3.1.1	Monitoring Direction, Reading a WAGO-I/O-PRO Variable ..	48
4.1.3.1.2	Monitoring Direction, Reading a Channel of the I/O Module..	49
4.1.3.1.3	Control Direction, Writing a WAGO-I/O-PRO Variable	50
4.1.3.1.4	Control Direction, Writing a Channel of the I/O Module.....	51

4.1.3.1.5	Monitoring Direction and Control Direction, Reading and Writing a WAGO-I/O-PRO Variable	52
4.1.3.2	“DataSet” Tab	53
4.1.3.3	“Reporting” Tab	54
4.1.3.4	“GOOSE” Tab	56
4.1.3.4.1	Creating a New “GOOSE Control Block” (GCB)	59
4.1.3.4.2	“General” Area	59
4.1.3.4.3	“Publisher” Area	59
4.1.3.4.4	“Subscriber Configuration” Area	60
4.1.4	Status Bar	61
4.1.4.1	Status Bar of a “Logical Device Object” (LDC)	61
4.1.4.2	Status Bar of a “Logical Name Class Object” (LNC)	61
4.1.4.3	Status Bar of a “Common Data Class Object” (CDC)	62
4.1.4.4	Status Bar of an Attribute	62
4.2	“Logical Name Classes” acc. IEC 61850 / IEC 61400-25	63
4.3	IEC 61850 Functionality	66
4.3.1	Project Planning and Design	67
	List of Figures	68
	List of Tables	70

1 Notes about this Documentation

1.1 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.2 Symbols

 **DANGER**

Personal Injury!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **DANGER**

Personal Injury Caused by Electric Current!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **WARNING**

Personal Injury!

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

 **CAUTION**

Personal Injury!

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE

Damage to Property Caused by Electrostatic Discharge (ESD)!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note

Important Note!

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.

Information

**Additional Information:**

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.3 Number Notation

Table 1: Number Notation

Number Code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.4 Font Conventions

Table 2: Font Conventions

Font Type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Program Files\WAGO Software</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualifications

All sequences implemented on WAGO-I/O-SYSTEM 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

All changes to the coupler or controller should always be carried out by qualified personnel with sufficient skills in PLC programming.

2.1.3 Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions

Fieldbus couplers, fieldbus controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to actuators or higher-level control systems. Using programmable controllers, the signals can also be (pre-) processed.

The devices have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the devices in wet and dusty environments is prohibited.

Operating the WAGO-I/O-SYSTEM 750 devices in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section “Device Description” > “Standards and Guidelines” in the manual for the used fieldbus coupler/controller.

Appropriate housing (per 94/9/EG) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

2.1.4 Technical Condition of Specified Devices

The devices to be supplied ex works are equipped with hardware and software configurations, which meet the individual application requirements. WAGO Kontakttechnik GmbH & Co. KG will be exempted from any liability in case of changes in hardware or software as well as to non-compliant usage of devices.

Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

2.2 Safety Advice (Precautions)

For installing and operating purposes of the relevant device to your system the following safety precautions shall be observed:



DANGER

Do not work on devices while energized!

All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.

DANGER

Install the device only in appropriate housings, cabinets or in electrical operation rooms!

The WAGO-I/O-SYSTEM 750 and its components are an open system. As such, install the system and its components exclusively in appropriate housings, cabinets or in electrical operation rooms. Allow access to such equipment and fixtures to authorized, qualified staff only by means of specific keys or tools.

NOTICE

Replace defective or damaged devices!

Replace defective or damaged device/module (e.g., in the event of deformed contacts), since the long-term functionality of device/module involved can no longer be ensured.

NOTICE

Protect the components against materials having seeping and insulating properties!

The components are not resistant to materials having seeping and insulating properties such as: aerosols, silicones and triglycerides (found in some hand creams). If you cannot exclude that such materials will appear in the component environment, then install the components in an enclosure being resistant to the above-mentioned materials. Clean tools and materials are imperative for handling devices/modules.

NOTICE

Clean only with permitted materials!

Clean soiled contacts using oil-free compressed air or with ethyl alcohol and leather cloths.

NOTICE**Do not use any contact spray!**

Do not use any contact spray. The spray may impair contact area functionality in connection with contamination.

NOTICE**Do not reverse the polarity of connection lines!**

Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

NOTICE**Avoid electrostatic discharge!**

The devices are equipped with electronic components that may be destroyed by electrostatic discharge when touched. Please observe the safety precautions against electrostatic discharge per DIN EN 61340-5-1/-3. When handling the devices, please ensure that environmental factors (personnel, work space and packaging) are properly grounded.

3 Application Description

3.1 Abbreviations

Table 3: Abbreviations

Abbreviation	Explanation	Description
CDC	Common Data Class	An LNC object is made up of many CDCs. A single CDC contains many attributes.
DA	Data Attribute	Individual attribute of a DO that typically consists of a variable and, if necessary, a trigger option.
DO	Data Object	Data object within an LN.
FC	Functional Constraint	Assigned to data attributes, a service (functional constraint) determines multiple items, such as what access rights and initializations are available. → IEC 61850
GCB	GOOSE Control Block	Compilation of data objects that can be sent by a publisher or received by a subscriber.
GGIO	Generic Process Input Output	A GGIO is a generic information object and can be used if no typical IEC 61850 information object is suitable for the given application.
GOOSE	Generic Object Oriented Substation Events	Transmission protocol for real-time multicast communication.
IEC 61850	International Electrotechnical Commission 61850	IEC 61850 is an international standard that defines a general transmission protocol. The protocol is used for protection, as well as for control systems in electrical substations with medium- and high-voltage technology.
IED	Intelligent Electronic Device	According to IEC 61850, an IED is an intelligent field device.
Internal data bus	Internal data bus	The internal data bus combines the I/O modules of the WAGO-I/O-SYSTEM 750 Series to exchange data.
LD	Logical Device	Logical device and the top node in the configuration structure.
LN	Logical Node	A node located under the LD. The node determines the data type provided.
LNC	Logical Name Class	LNC is the general term for all IEC 61850 information object types, e.g., GGIO.
LLN0	Logical Node Zero	Every IED receives one LLN0 information object.
LPHD1	Physical Device Information	Every IED receives exactly one LPHD1 information object.
MMS	Manufacturing Messaging Specification	Transmission protocol according to IEC 61850-8-1 with server/client communication via TCP/IP.
SCL	Substation Configuration Language	SCL is an XML-based language used as an exchange format for IEC 61850 engineering data.

3.2 Tasks

This quick start guide describes IEC 61850-compliant methods to:

- Create a server with WAGO controls.
- Establish GOOSE communication with WAGO controls.

The IEC 61850 Configurator (V2.00.23 or higher) integrated into WAGO-I/O-PRO is the configuration program. To check operation, a connection must be established between the controller and PC.

Server and GOOSE communication should contain a GGIO Logical Node Object with both a binary and analog signal.

A fieldbus controller for telecontrol systems (see section “Hardware Requirements”) is connected to the IEC 61850 client “IEDScout” from OMICRON and read.

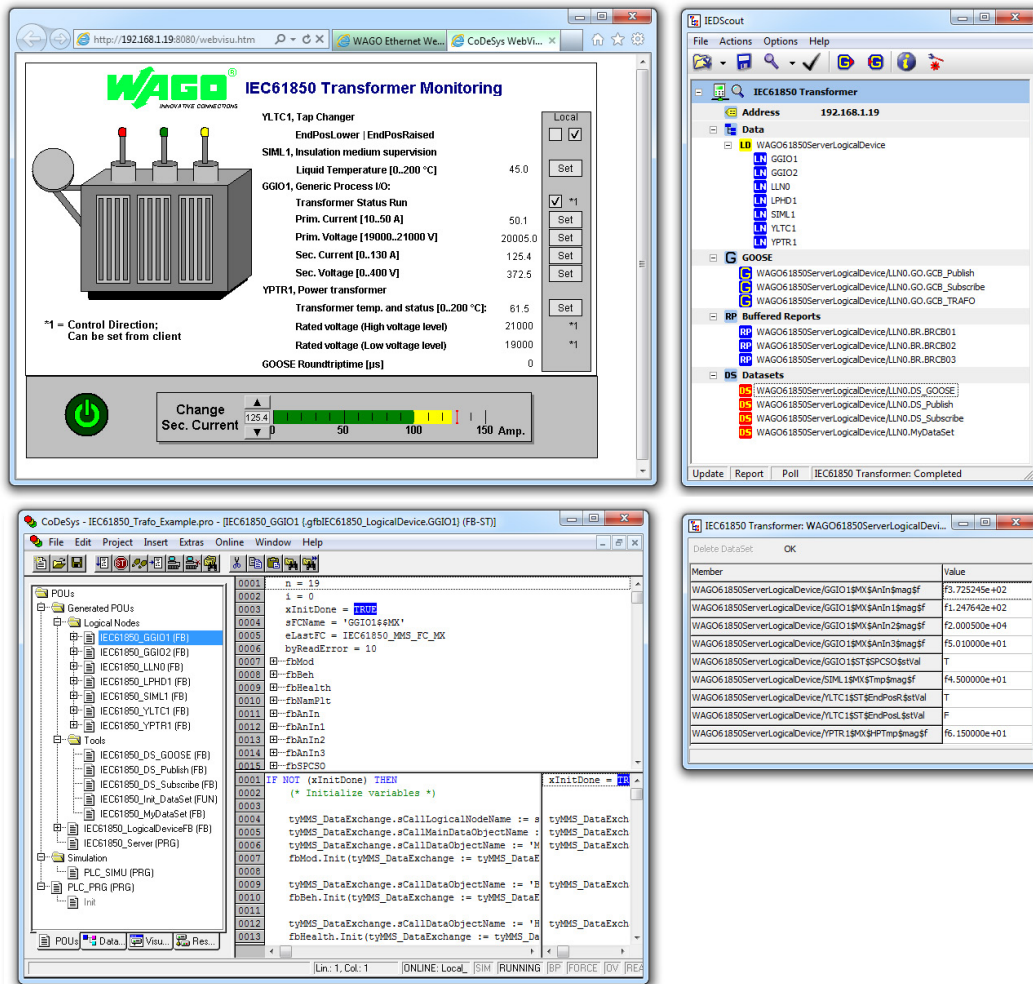


Figure 1: Example of Transformer Monitoring Application with Connected IEC 61850 Client

3.3 Hardware Requirements

You can use the following WAGO hardware components:

Table 4: Hardware Requirements

Hardware components	Item No.	Release version
Programmable fieldbus controller for telecontrol applications	750-872	Release version (05) or higher
Programmable ETHERNET controller for telecontrol applications	750-880/xxx-001	Release version (05) or higher
Programmable ETHERNET controller for telecontrol applications	750-880/xxx-002	Release version (05) or higher
I/O-IPC-C6 Linux [®] TeleControl	758-874/000-130	Release version (09) or higher
I/O-IPC-C6 Linux [®] TeleControl PDP master	758-874/000-131	Release version (09) or higher
PFC200 CS 2ETH RS TeleControl/T	750-8202/025-001	Release version (02) or higher
PFC200 CS 2ETH RS TeleControl ECO/T	750-8202/025-002	Release version (02) or higher
PFC200 CS 2ETH RS CAN DPS TeleControl/T	750-8206/025-001	Release version (03) or higher

3.4 Software Requirements

- WAGO-I/O-PRO v2.3.9.46 or higher
- IEC 61850 client, e.g., the “IEDScout” product from OMICRON (www.omicron.at)

3.5 Requirements and Capabilities

Commission and program WAGO-I/O-SYSTEM fieldbus controllers using WAGO-I/O-PRO v2.3.

Information



More information for a quick start!

For information on commissioning and programming a WAGO Fieldbus Controller, please download the quick start guide for the ETHERNET TCP/IP (750-841) programmable fieldbus controller. You will find it at www.wago.com.

3.6 MMS Communication

3.6.1 Create New Project and Open the IEC 61850 IEC 61850 Configurator

1. Launch WAGO-I/O-PRO.
2. Create a new project. Click **File > New**.

The “Target System Settings” dialog appears.

3. Under **Configuration**, select the hardware you are using.
4. Click on **[OK]**.

The “New Module” dialog appears.

5. Click the **[Cancel]** button because you do not want to create a new module.
6. In the main menu, select the **Resources** tab.
7. In the directory, select the **PLC Configuration** entry.

The “PLC Configuration” workspace appears.

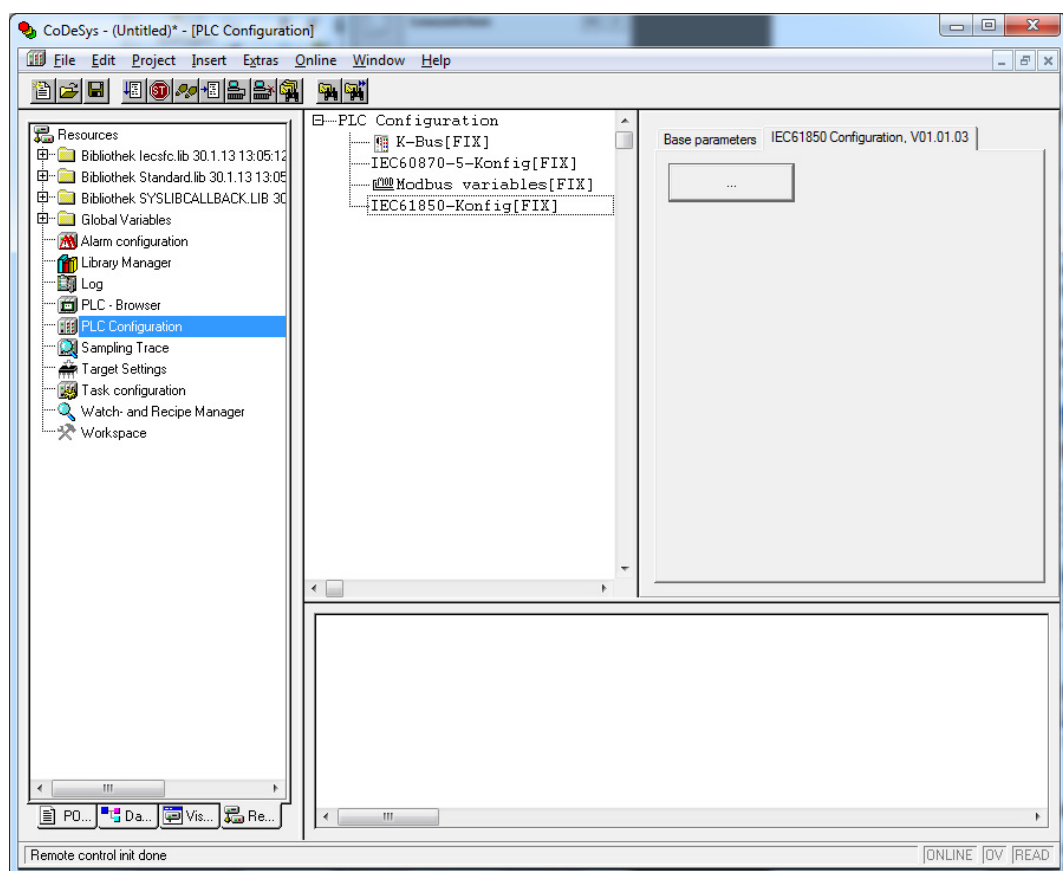


Figure 2: “PLC Configuration” workspace

8. Select the **IEC61850-Config(FIX)** entry in the tree structure.

The “WAGOIEC61850 Konfigurator” dialog appears.

3.6.2 Create an IEC 61850 Configuration

1. Click the [>>] button below the server to create a new **LogicalDevice**.

Both required LNC instances (“LLN0” and “LPHD1”) are also created.

2. Then select the LogicalDevice object and create a new “GGIO” type object instance by clicking the [>>] button below the LogicalDevice object.

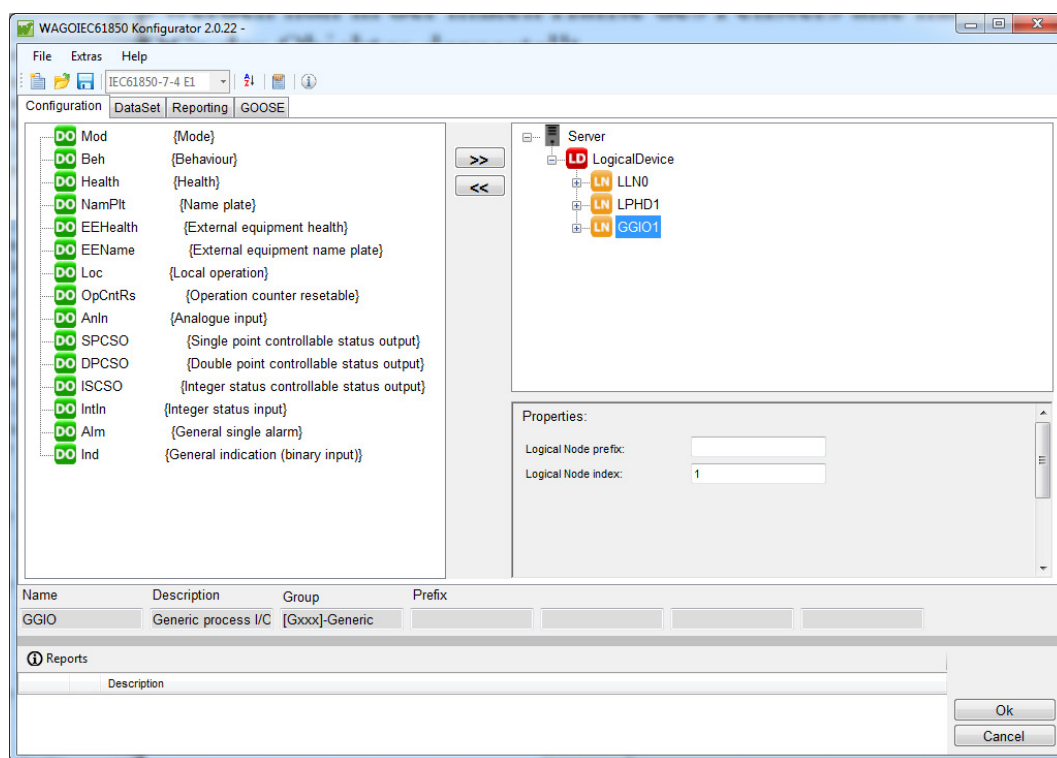


Figure 3: IED configuration, IED expanded by one LNC instance of the “GGIO” type

3. Select the object “GGIO 1”.

All CDC types (*mandatory* and *optional*) are displayed on the left half of the window.

4. Select the “AnIn” and “Ind” entries on the left side and click [>>] to apply.

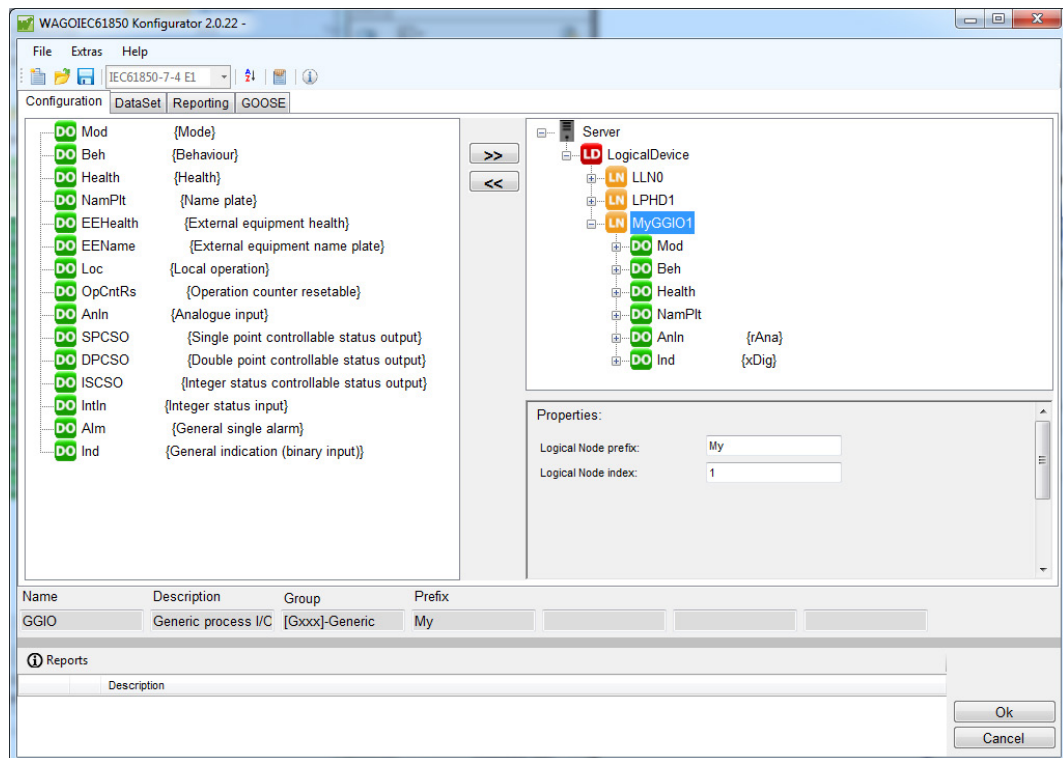


Figure 4: IED configuration, generate additional CDC instance

5. Link the IEC 61850 attributes with the WAGO-I/O-PRO variables.
In this example:
AnIn.mag with “rAna”
Ind.stVal with “xDig”
6. If necessary, you can define an initial value in the input field to the right of the variable name.
Activate the “Autom. declare” option and the variable from the IEC 61850 Configurator is declared as a global WAGO-I/O-PRO variable.

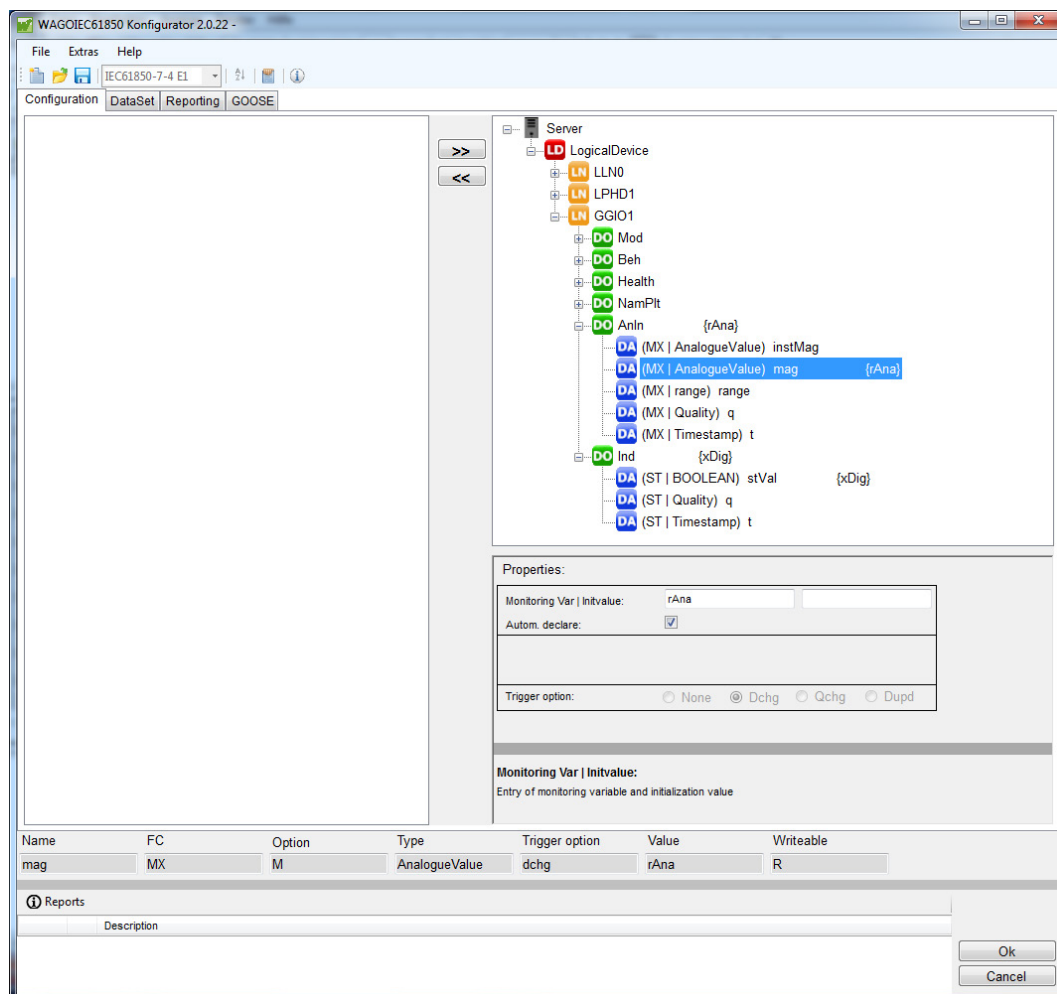


Figure 5: IED configuration, monitoring direction: linking IEC 61850 attributes to WAGO-I/O-PRO variable

Note



Deactivate “Autom. declare” option for direct connections!

If you establish a direct connection from an I/O module pin to an IEC 61850 attribute by using the same variable name, deactivate the “Autom. declare” option.

Should this option remain active, the variable is declared by the internal data bus configurator and as the PLC program is compiled, the following error message appears: “The variable is declared twice.”

In the following step, you will create a “dataset” according to the IEC 61850 configuration defined here. A dataset is a compiled group of IEC61850 configuration data.

3.6.3 Create Dataset

1. To create a dataset, navigate to the **DataSet** tab in the Configurator.
2. Press the **[New]** button to create a new dataset. You can rename the dataset in the “Name” input field.
3. Assign the following attributes to the dataset:
 “Server.LogicalDevice.GGIO1.AnIn.mag” and
 “Server.LogicalDevice.GGIO1.Ind.stVal.”
 To do this, select the attributes on the left-hand side and press the **[>>]** button to assign them to the relevant dataset.

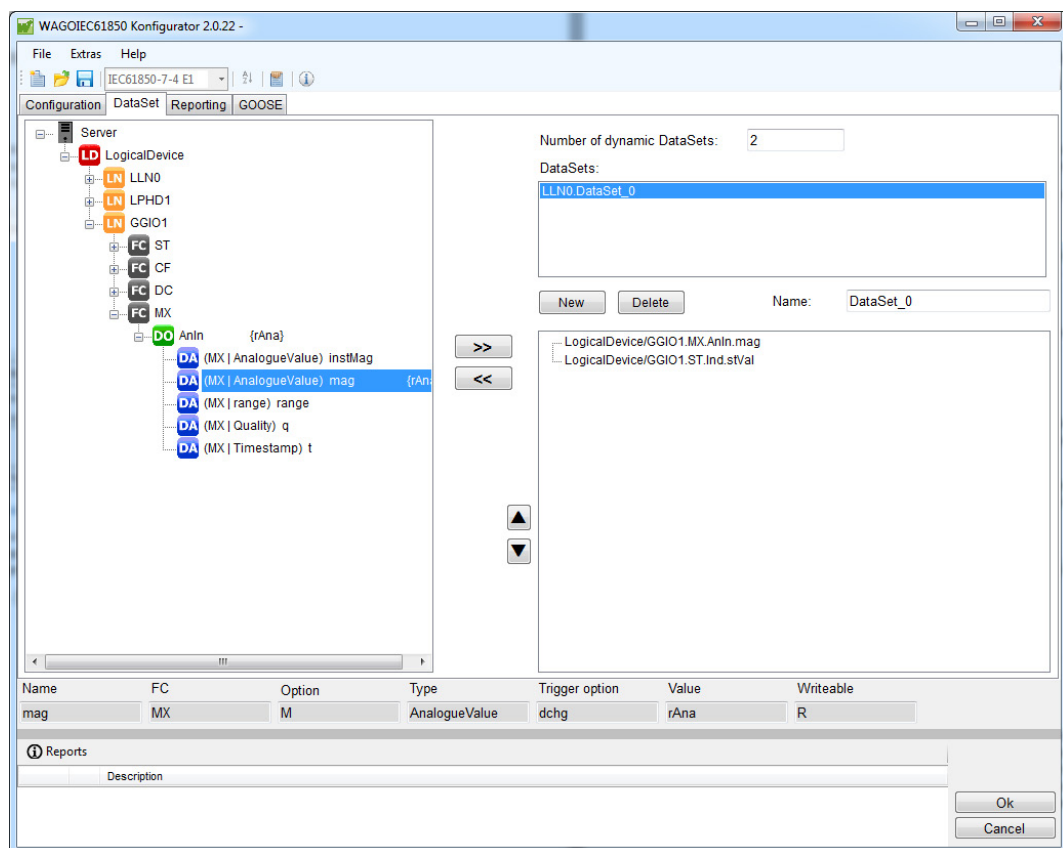


Figure 6: Dataset configuration

Note



Note the trigger option in the status bar!

Please note the trigger option (“Trigger option”) of the attribute defined in the IEC 61850 specification. The trigger option determines whether the report (see below) assigned to the dataset is sent when the value of this attribute changes.

Information



Additional information about the trigger options!

Additional in-depth information about the various trigger options can be found in the status bar description in the appendix.

In the following step, you will assign a report to the dataset defined here. A report transports the data assigned to it as a dataset to the connected client during a triggering event.

The maximum number of configurable data sets, data objects and data attributes is not limited in a program. However, the maximum usable number is limited by the memory and computing power of the fieldbus controller/IPC.

3.6.4 Create a Report

1. To create a report, navigate to the **Reporting** tab in the Configurator.
2. Press the **[New]** button to create a new report. You can rename the report in the “Name” input field.
3. Using the “DataSet” selection list, assign a dataset to the created report. The functions of the individual parameters are described in the appendix.

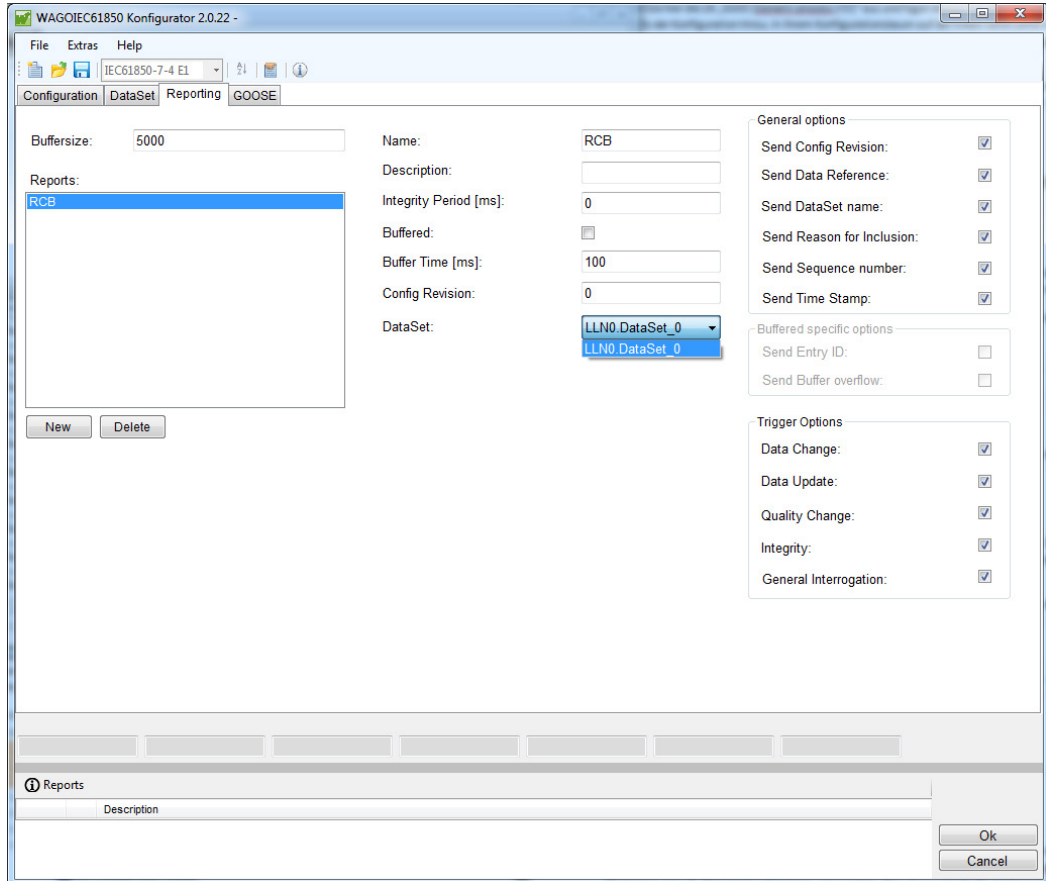


Figure 7: Reporting configuration

3.6.5 Generate Code, Translate and Load Solution into the Controller

1. Press the **[Ok]** button to apply changes made using the IEC 61850 Configurator.
Press **[Cancel]** to discard the changes.

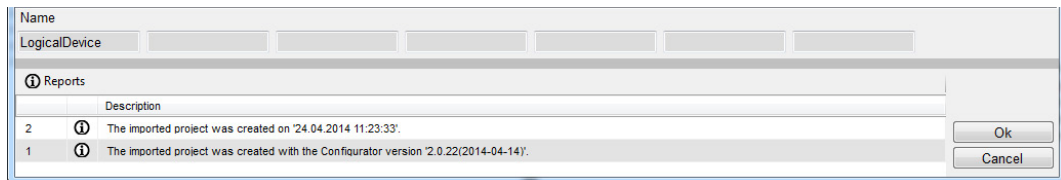


Figure 8: Status bar buttons to apply or discard changes

2. Close the Configurator.

WAGO-I/O-PRO generates the code. The generated IEC 61850 solution is saved in the “Generated POU’s” folder.

3. To compile the code, select the **Project > Compile** menu item.

A completed example is shown in the figure below.

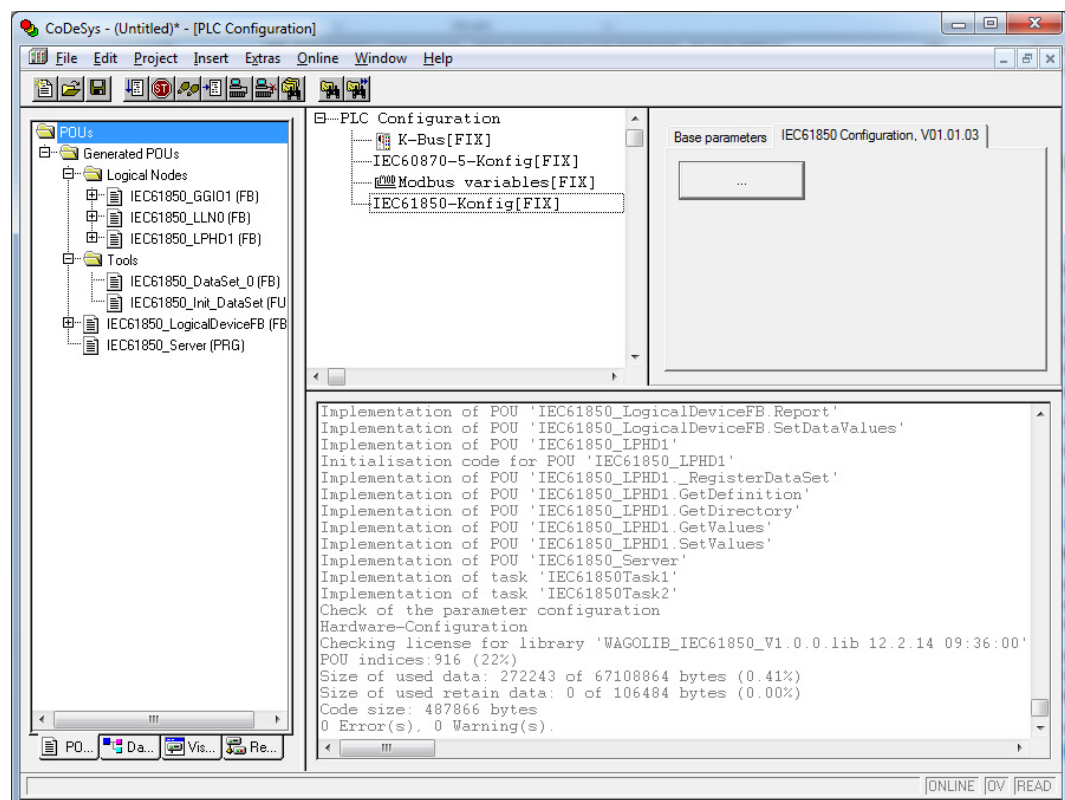


Figure 9: Generated and compiled IEC 61850 solution in WAGO-I/O-PRO

4. Load the solution into the controller via the **Online > Login** menu item.
5. Go to **Online > Start** to start the controller.

Note



Online communication requires a WAGO-I/O-PRO gateway configuration!
To meet the requirements for online communication between WAGO-I/O-PRO and the controller, please ensure that you have configured the WAGO-I/O-PRO gateway under **Online > Communication parameters....**

As soon as the IEC 61850 solution is running in the WAGO controller, you can access it with an IEC 61850 client (“IEDScout” from OMICRON is used here).

3.6.6 Connect WAGO Controller with IEC 61850 Client

1. Connect the WAGO controller with an IEC 61850 client.
In this example, use the “IEDScout” software as part of OMICRON's “IEC 61850 Package”.
2. Install and launch “IEDScout,” then configure the connection to the required IEC 61850 server.
To do this, select the **Actions > Configure** menu item.
3. In the window that opens next, press **[New]** to create a new server. Assign a suitable name to the server, e.g., “WAGO61850Server”, then enter the controller's IP address controller in the input field.
4. Read the successfully configured and connected server in the “IEDScout”.
To do this, select the **Actions > Discover** menu item.
The progress is displayed in a separate window. Once finished, the “IEDScout” displays the object structure, datasets and report from the IEC 61850 server.
5. Double-click the data set. A new window appears that displays the current data from the defined report.

Note



Requirement for cyclical data!

Please note that the attribute “\$LLN0\$BR\$RCB_0\$RptAna” must be “TRUE” after a successful connection for cyclically updating the data by the “IEDScout”. If you do not have access to a full version, or you would like to simplify the process, activate the **[Permanent Report Activation]** switch in the option dialog of the WAGO IEC 61850 Configurator for test purposes only.

Note: This setting is only used for testing and does not conform to any standard!
Undo this setting in production systems!

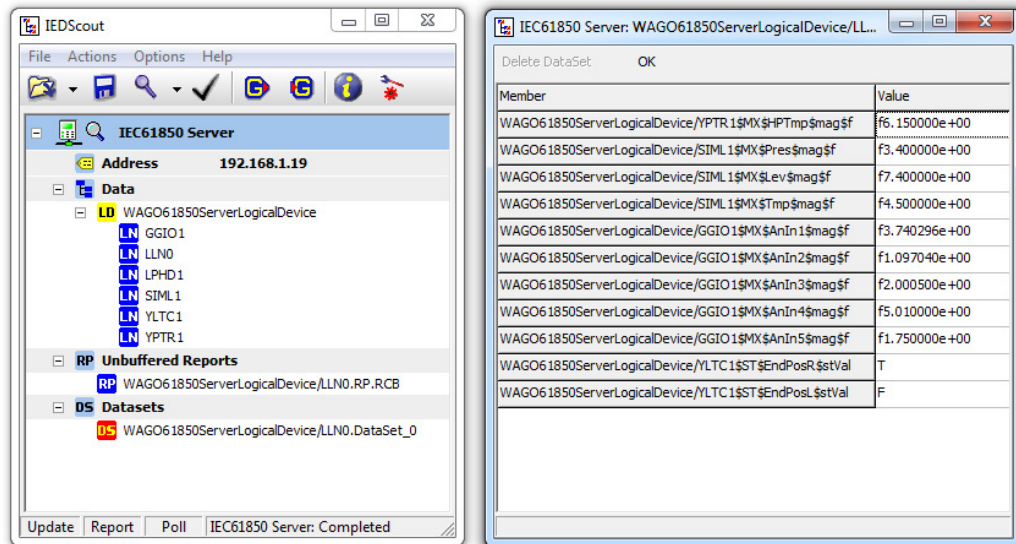


Figure 10: “IEDScout” with object structure and dataset connected to WAGO controller

Note



TCP/IP communication can be recorded!

Please note that TCP/IP communications between the OMICRON Client and WAGO server can be recorded using a protocol analyzer, e.g., “Wireshark” (formerly “Etherreal”). “Wireshark” includes an IEC 61850 plug-in that can directly decode communication.

3.6.7 Conclusion

You have created an IEC 61850 server using a WAGO controller and the IEC 61850 Configurator that is integrated into *WAGO-I/O-PRO*.

This server has a GGIO Logical Node Object that contains both binary and analog signals. The control system can be connected and read out using “IEDScout,” an IEC 61850 client from OMICRON.

Note



Execute a connectivity test!

Perform a connectivity test, e.g., via reports with activated “Integrity Period [ms]” option (gradual general query). If you set a value of 5000 ms, a report is sent to the clients every five seconds and the connection is tested.

With each telegram sent, the IEC 61850 server verifies connection to the client. This makes it necessary to send telegrams.

If a connectivity test has not been performed, the connection is blocked after an outage and the client is not able to re-connect.

Note the following points when “Control” objects are written by a client:

- “Data Attributes” that can be written by a client are indicated in the Configurator by the “RW” (Read/Write) access authorization.
- The “LLN0\$ST\$Mod\$stVal” attribute must be set to “ON” (1). Otherwise, reports are not sent and “Control” values cannot be transferred. These values are of the **Enumeration** type. The corresponding reference is found in the IEC 61850-7-4.6 “Data semantics (Mode).”
- Some “Control” values can only be written as a structure. These include values with the “Function Constraint” (CO), e.g., “LogicalNode\$CO\$DataObject\$Oper”.
- If the “Control” values involve **Enum** type variables, note that these variables can only be written in the range valid for them.

3.7 GOOSE Communication

The following example outlines the creation of GOOSE cross-communication between two GOOSE-compatible WAGO devices. You can use the following devices:

- IPCs for telecontrol technology
- PFC200 for telecontrol technology
- Programmable fieldbus controllers for telecontrol technology (750-880/xxx-001 or 750-880/xxx-002 with release version 07 or higher)

One device is used as a publisher (750-880/0xx-00x with release version 07) and sends GOOSE telegrams.

3.7.1 Configure Publisher

1. Launch *WAGO-I/O-PRO* and select **File > New** to create a new project.
2. In the **Target System Settings** dialog, select the relevant device (e.g., the entry, “WAGO_750-880-xxx-001_ -_750-880/040-001” for the programmable fieldbus controller 750-880/025-001) and click **[OK]** to confirm.
3. Click **[Cancel]** to close the dialog that opens. This dialog creates a program for automation applications. It is not required here.
4. In the main menu, click the **Resources** tab.
5. In the directory, select the **PLC Configuration** entry.

The “PLC Configuration” workspace appears.

6. In the tree structure, select the **IEC-61850-Konfig[FIX]** entry.
7. Select the **IEC 61850 Configuration, V01.01.03** tab.
8. Click the **[...]** button.

The “WAGOIEC61850 Configurator 2.x.x” workspace appears.

9. In the main menu, click the **Configuration** tab.

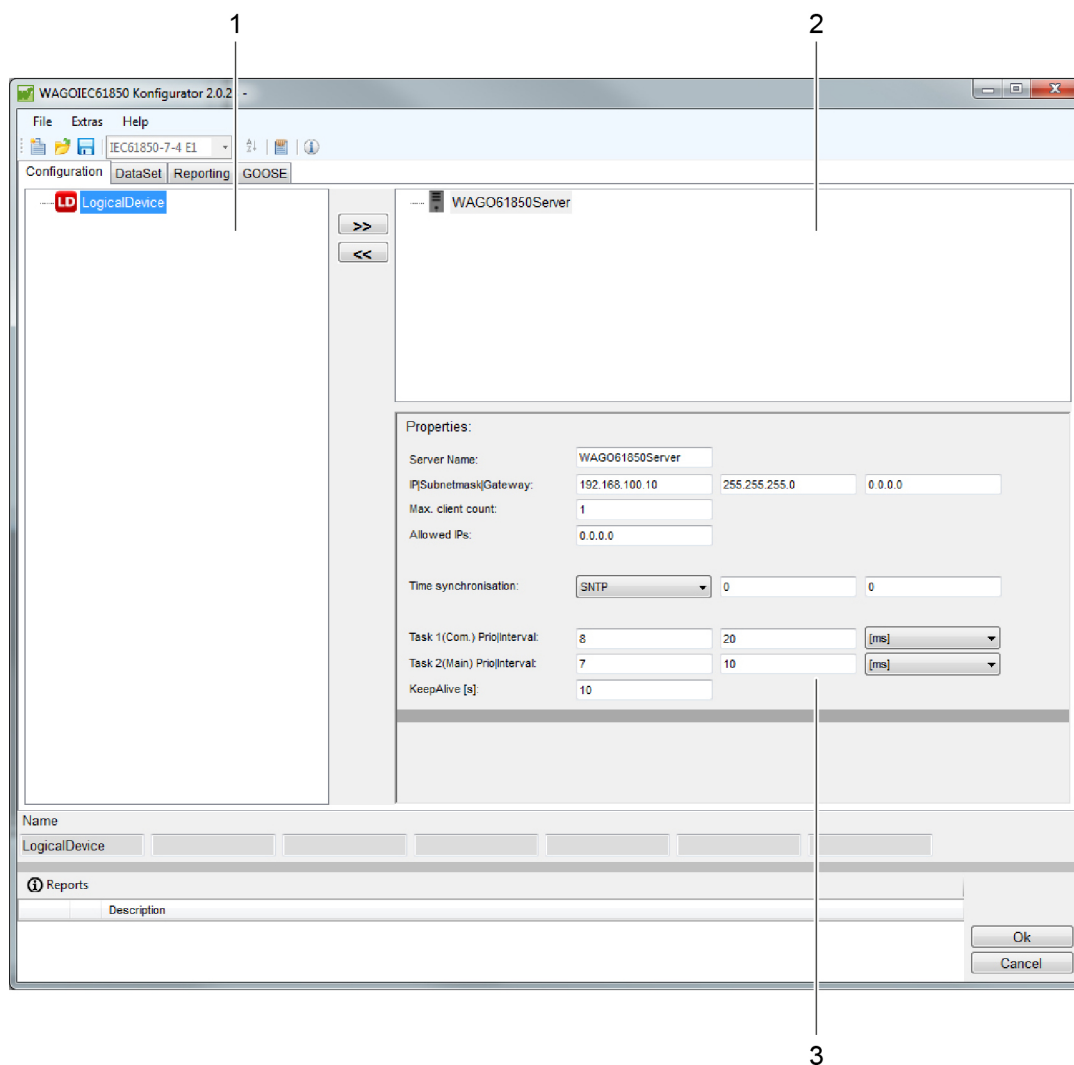


Figure 11: “Configuration” tab

Table 5: Key for “Configuration tab” figure

Pos.	Description
1	Selection area
2	Configuration area
3	Properties area

10. Click the **LogicalDevice** entry in the selection area.
11. Click the [>>] button to apply the entry in the configuration area.
12. Click the **LogicalDevice** entry in the configuration area.

A list of available LDs will appear in the selection area.

13. In the directory, click the [+] button in front of the [Gxxx] **Generic** entry.

A tree structure with subentries appears.

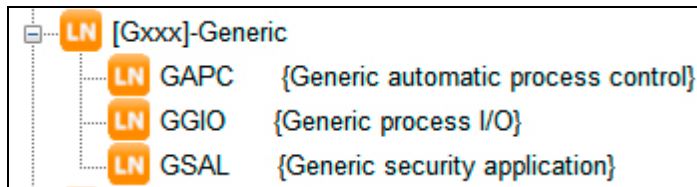


Figure 12: Generic logical nodes

14. Click the **GGIO** {Generic process I/O} entry.
15. Click the [>>] button to apply the entry in the configuration area.
16. Click the **GGIO1** entry in the configuration area.

A list of available DOs appears in the selection area.

17. Click the **AnIn** {Analog input} entry in the selection area.
18. Click the [>>] button to apply the entry in the configuration area.
19. Click the [+] button in front of the **AnIn** entry in the configuration area.

A list of available DAs appears in the configuration area.

20. Select the **(MX | AnalogValue) mag** entry in the configuration area.
21. Make the following settings in the properties area:
 - Enter the **rAnValue** variable name in the “Monitoring Var.” input field.
 - Select the “Autom. declare” checkbox.

Figure 13: Variable assignment for the analog input

22. Click the **GGIO1** entry in the configuration area.

A list of available DOs appears in the selection area.

23. Click the **Ind {General indication (binary input)}** entry in the selection area.
24. Click the [>>] button to apply the entry in the configuration area.
25. Click the [+] button in front of the **Ind** entry in the configuration area.

A list of available DAs appears in the configuration area.

26. Select the **(ST | BOOLEAN) stVal** entry in the configuration area.
27. Make the following settings in the properties area:
 - Enter the **xBinValue** variable name in the “Monitoring Var.” input field.
 - Select the “Autom. declare” checkbox.
28. In the main menu, click the **DataSet** tab.

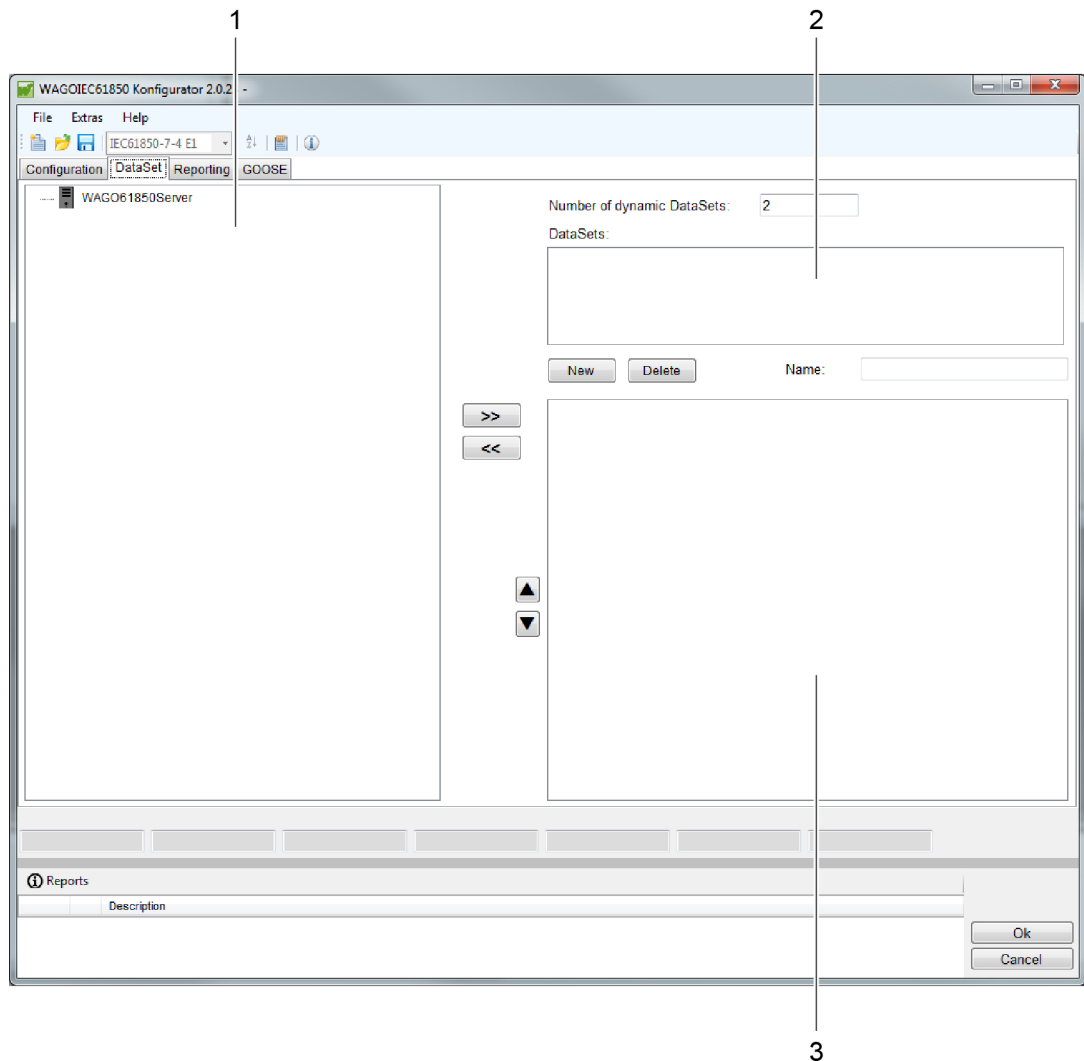


Figure 14: “DataSet” tab

Table 6: Key for “DataSet tab” figure

Pos.	Description
1	Selection area
2	Configuration area (create dataset)
3	Configuration area (configuration dataset)

29. Click the **[New]** button in the configuration area (create dataset).

A new dataset is created.

30. Click the **[+]** button in front of the **MX** entry in the selection area.

31. Click the **[+]** button in front of the **AnIn** entry in the selection area.

A list of available DAs appears in the selection area.

32. Click the **(MX | AnalogValue) mag** entry in the selection area.

33. Click the **[>>]** button to assign the entry to the selected dataset.

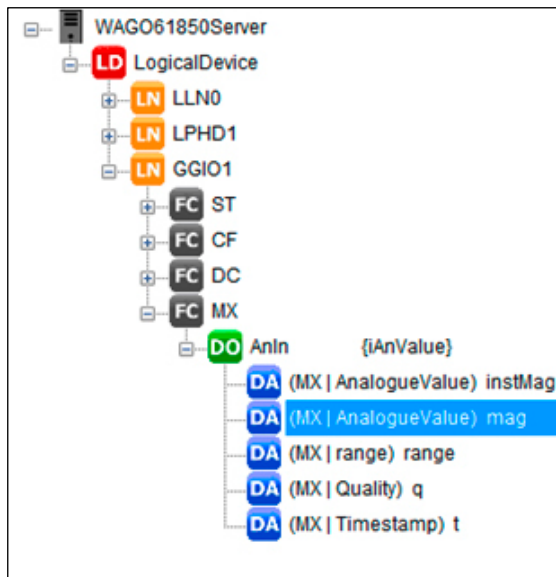


Figure 15: Configure dataset

34. Repeat steps 30–33 for the **(ST | BOOLEAN) stVal** entry.
In step 31, select the **Ind** entry.
35. In the main menu, click the **GOOSE** tab.

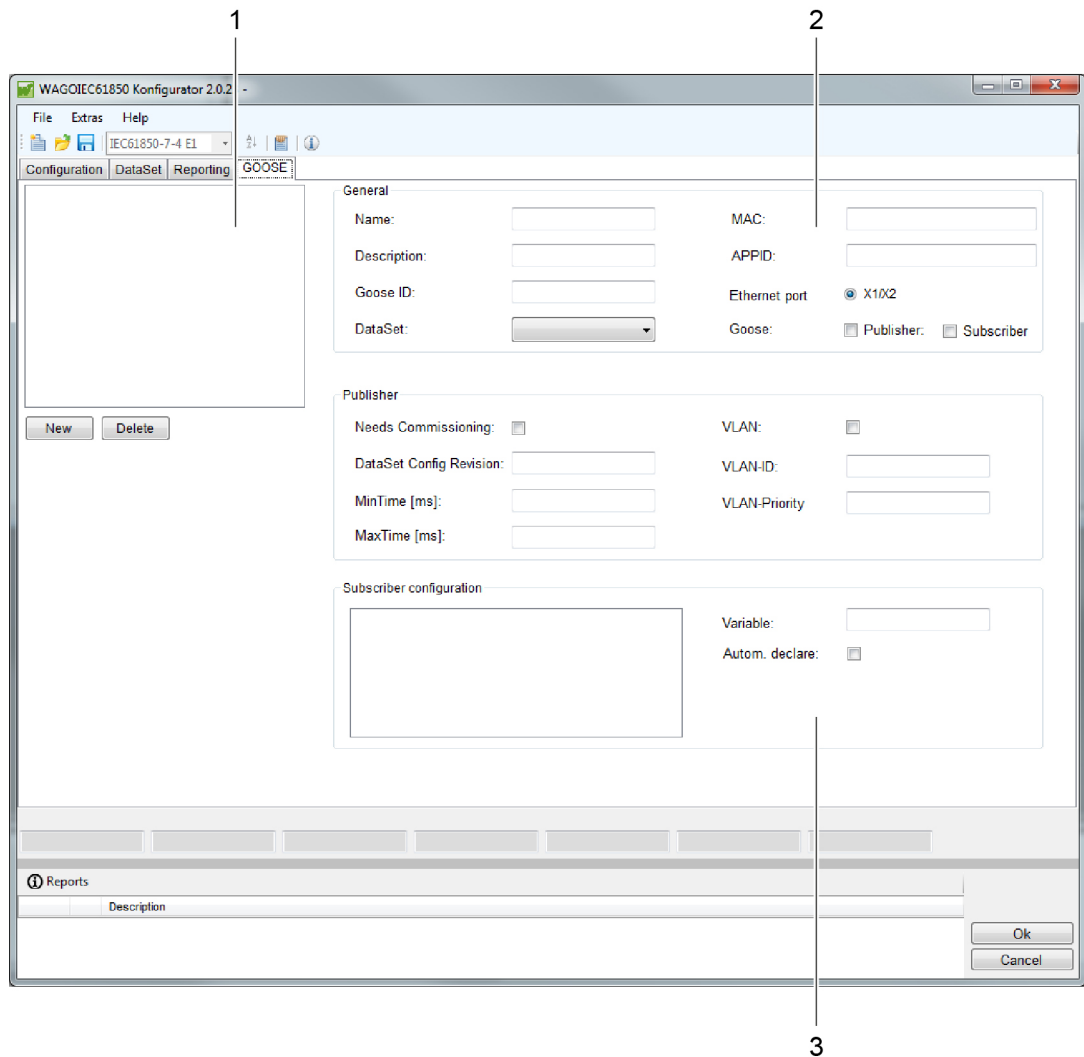


Figure 16: “GOOSE” tab

Table 7: Key for “GOOSE tab” figure

Pos.	Description
1	Configuration area
2	“General” area
3	“Subscriber Configuration” area

36. Click the **[New]** button in the configuration area.

A new “GOOSE Control Block” (GCB) is created.

37. Click the **DataSet** selection list in the “General” area.

38. Select the dataset previously created.

General			
Name:	<input type="text" value="GCB_0"/>	MAC:	<input type="text" value="01-0C-CD-01-00-00"/>
Description:	<input type="text"/>	APPID:	<input type="text" value="0"/>
Goose ID:	<input type="text" value="GoCBRef"/>	Ethernet port	<input checked="" type="radio"/> X1/X2
DataSet:	<input type="text"/>	Goose:	<input checked="" type="checkbox"/> Publisher: <input type="checkbox"/> Subscriber

Figure 17: General GOOSE settings for a PFC200

39. Click on the **[OK]** button.

The publisher has been configured.

3.7.2 Configure Subscriber

All steps to be performed are very similar to those used to configure the publisher. The only difference is that specific parameters must be set in the “GOOSE” tab.

1. Repeat steps 1–34 in the section “Configure Publisher”.
2. In the main menu, click the **GOOSE** tab.
3. Click the **[New]** button in the configuration area.

A new GCB is created.

4. Select the **Subscriber** checkbox in the “General” area.
5. Click the **DataSet** selection list in the “General” area.
6. Select the first data set.
7. In the “Subscriber Configuration” area, click in the **Variable** input field.
8. Enter the CODESYS variable **rAnSubscr** in the input field.
9. Select the **Autom. declare** checkbox in the “Subscriber Configuration” area.

The CODESYS variable **rAnSubscr** is assigned to the data attribute.

10. Repeat steps 4–8 for the second data attribute. Assign the CODESYS variable **xBinSubscr** to this data attribute.

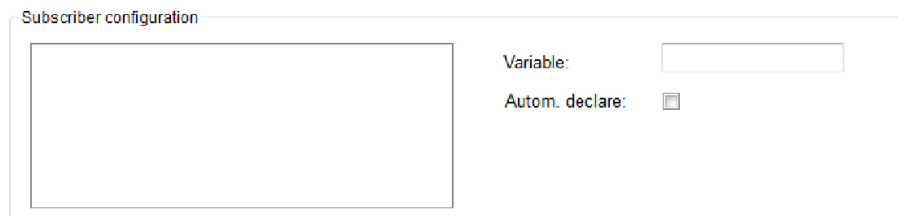


Figure 18: Subscriber configuration

11. Click on the **[OK]** button.

The subscriber has been configured.

3.7.3 Commission the Configuration

When both configured applications are loaded in the GOOSE device and started, the publisher sends GOOSE telegrams. A “network sniffer program”, e.g., “Wireshark” can record and view the telegrams.

PC and software must be in the same network as the two GOOSE devices. Port mirroring does not have to be set up as switches can distribute the GOOSE telegrams to all connected nodes (multicast).

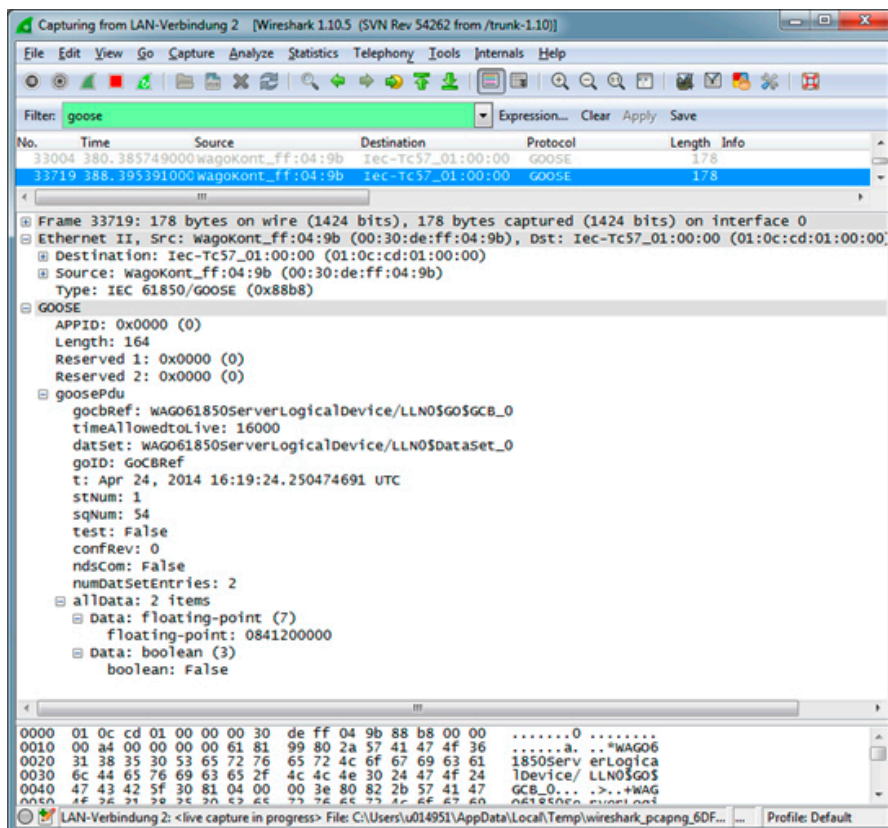


Figure 19: GOOSE telegram in Wireshark

The transferred measured value and the binary value are identified within the message telegram. To check communication, both CODESYS projects can be opened simultaneously. A value can be set for the **rAnValue** variable in the publisher. This value is transmitted immediately and echoed in the data range of the captured packets. The time interval between the telegrams always doubles until the **MaxTime** interval is reached. In this time interval, the telegrams are transmitted cyclically.

It should then be determined if the telegrams are received and correctly evaluated by other communication nodes. The global variable **rAnSubscr** can be viewed in the second CODESYS instance (subscriber).

In debug mode for the CODESYS instance of the publisher, you can write a value to a variable.

1. Double-click the selected variable.
2. Click the “Resources” entry in the tree structure.
3. Click the [+] button in front of the “Global Variables” entry.
4. Click the “IEC 61850 Generated Global_Variables (CONSTANT)” entry.

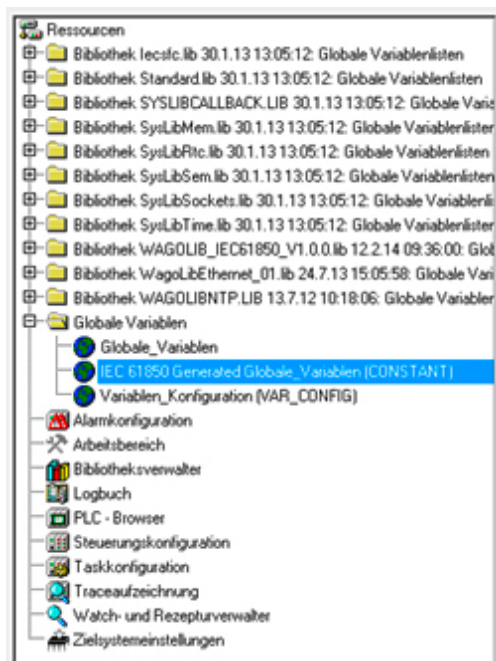


Figure 20: Global variables

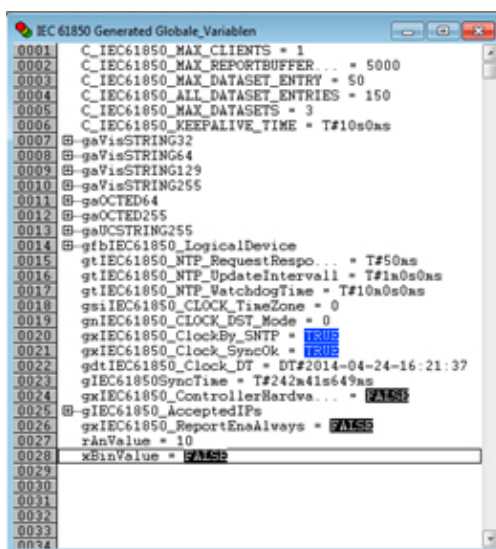


Figure 21: Publisher global variables

5. Double-click the **rAnValue** variable.

The “Write Variabler rAnValue” window appears.

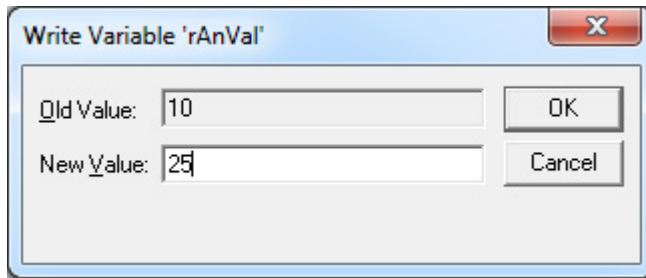


Figure 22: "Write Variable rAnValue" window

6. Enter any value in the "New Value" input field. This value may not be equal to the value entered in the "Old Value" input field.
7. Click the [OK] button.
8. Press the [Ctrl] and [F7] keys.

The new value is written to the variable.

```

gxIEC61850_ReportEnaAlways = FALSE
rAnValue = 10 < := 11 >
xBinValue = FALSE

```

Figure 23: Writting a variable value

9. Switch to the CODESYS instance of the subscriber.
10. Open the global variables there.

The value of the **rAnSubscr** variable should now correspond to the **rAnValue** value. If the value has changed according to the default value in Publisher, the transmission is working correctly.

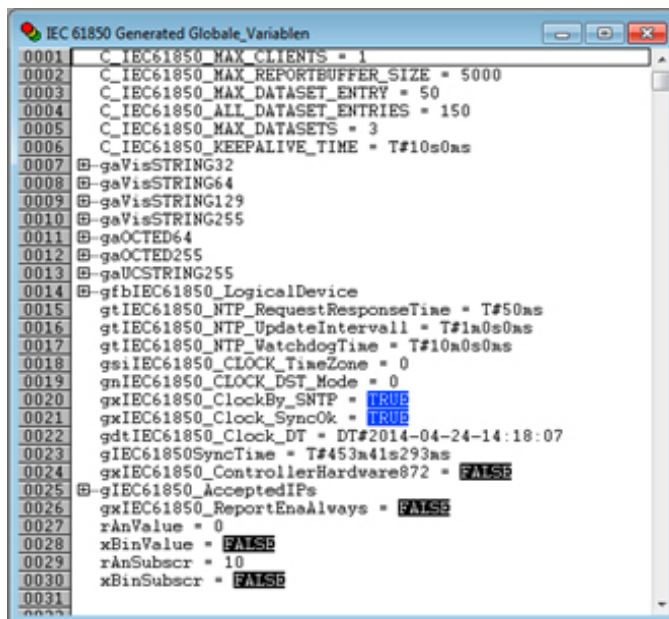


Figure 24: Global variables of the subscriber

4 Appendix

4.1 IEC 61850 Configurator

4.1.1 Overview

Different aspects of the IEC 61850 Configurator are explained below. Each aspect is described in a separate section.

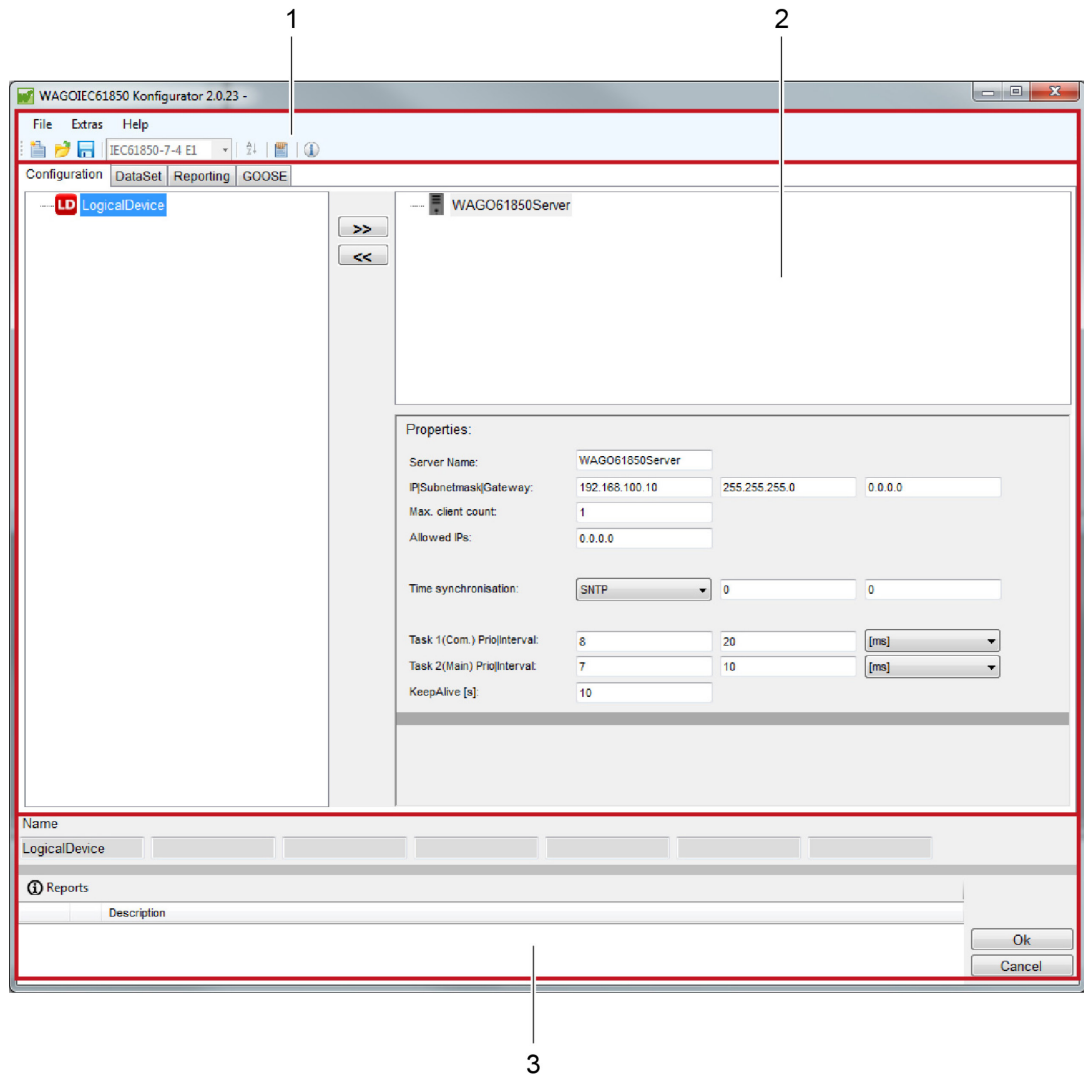


Figure 25: IEC 61850 Configurator overview

Table 8: Key for IEC 61850 Configurator overview

Number	Range	Description
1	Menu Bar	Using the functions in the menu bar, you can export and import Configurator data in WAGO XML and IEC 61850 SCL format. In addition, you can adjust the Configurator's display.
2	Operating range	In the individual workspace tabs, you can configure the object structure, datasets and IED reports.
3	Status bar	Buttons to close the dialog and for type-specific information on each selected object can be found in the status bar.

4.1.2 Menu Bar

4.1.2.1 “File” Menu

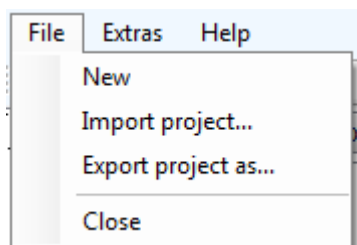


Figure 26: Configurator “File” menu

Table 9: Configurator “File” menu

Menu Item	Action
New	Discard current and create new IEC 61850 configuration.
Import project ...	Discard current and import new configuration. <ul style="list-style-type: none"> • “XML Files” format filter for proprietary WAGO 61850 Configurator format with all private data, e.g., WAGO-I/O-PRO MapKanal variables. • “SCL Files” format filter for IEC 61850 format to import data from other IEC 61850 tools.
Export project as ...	Exports current IEC 61850 configuration. <ul style="list-style-type: none"> • “XML Files” format filter for proprietary WAGO 61850 Configurator format with all private data, e.g., WAGO-I/O-PRO MapKanal variables. • “ICD File” format filter for IEC 61850 format to export data to other IEC 61850 tools.
Close	Closes the Configurator. If changes were made, a security question will appear asking if the affected information should be saved or discarded.

4.1.2.2 “Tools” Menu

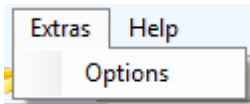


Figure 27: Configurator “Tools” menu

Table 10: Configurator “Tools” menu

Menu Item	Action
Options	Adjusts various Configurator display options.

“Options” dialog

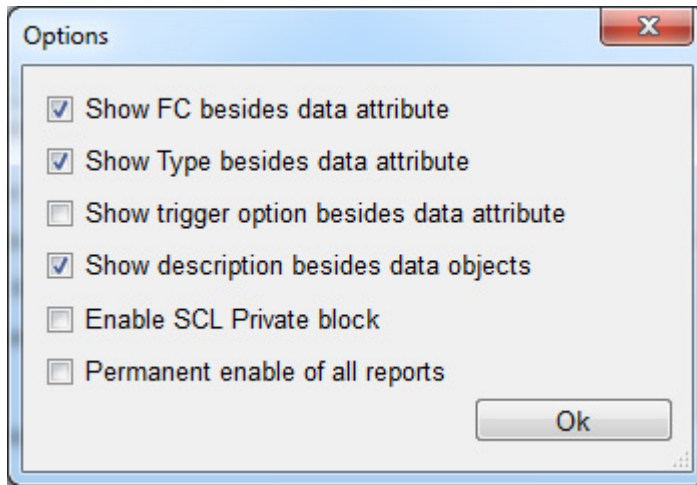


Figure 28: “Options” dialog

Table 11: Key for “Options dialog” figure

Option	Description
Show FC besides data attribute	Display option, shows the functional constraint of the attribute as a comment.
Show Type besides data attribute	Display option, shows the type of attribute as a comment.
Show trigger option besides data attribute	Display option, shows the trigger option of attribute as a comment.
Show description besides data objects	Display option, shows the description of attribute as a comment.
Enable SCL Private block	Ensures that shown CODESYS variables are also saved in SCL files (in the private area). This option is disabled by default because some third-party devices have problems reading in private data.
Permanent enable of all reports	Immediately activates sending of all reports without having to be activated by the client. This option is very useful for testing, especially to get an update on all reports in a given client. This option is only intended for test purposes and does not comply with any standard. It must be disabled in production systems.

Note



“Permanent Report Activation” option only to be used for testing purposes! Only enable the “Permanent Report Activation” option for testing and not during regular operation! In regular operation, no new connection is established after the connection is interrupted and the CODESYS application must be restarted!

4.1.2.3 “Help” Menu

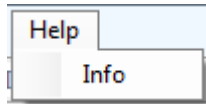


Figure 29: Configurator “Help” menu

Table 12: Configurator “Help” menu

Menu Item	Action
Info	Shows Configurator and library version information.

4.1.3 Operating range

4.1.3.1 “Configuration” Tab

In the “Configuration” tab, you can create, configure and parameterize an IED from existing LNC and CDC types.

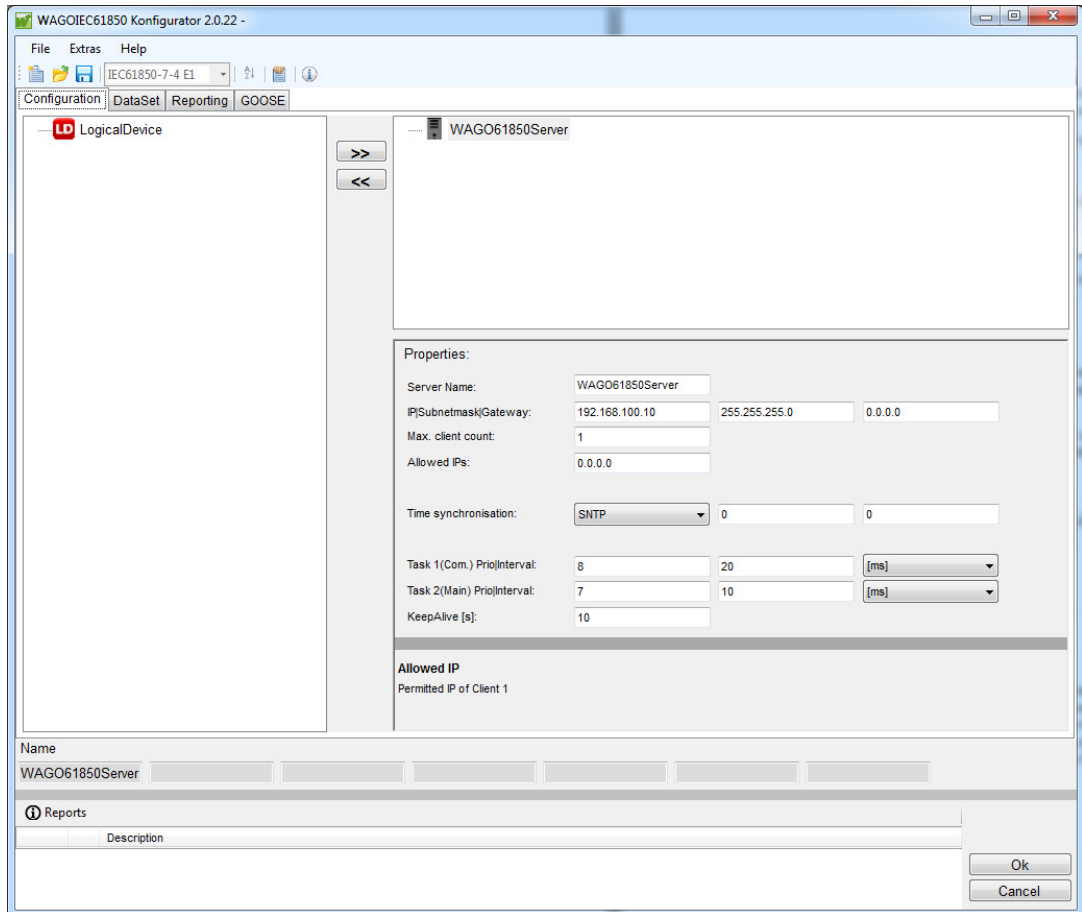


Figure 30: “Configuration” tab

The tab offers the following parameterization options:

Table 13: “Configuration” tab parameterization options

Setting Options	Description
Server Name	A name for the IEC 61850 server can be assigned (identifier).
IP	IP address of the IED. This setting is only relevant when exporting as an SCL file.
Subnet mask	If the subnet mask differs from the default value 255.255.255.0, it must be set offline in the Configurator and online in the controller. The Web-based management system can make this change. This setting is only relevant for exporting as an SCL file.
Gateway	If the gateway address differs from the default value 0.0.0.0, it must be set offline in the Configurator and also online in the controller. The Web-based management system can make this change. This setting is only relevant for exporting as an SCL file.
Max. client count	The maximum number of clients that can connect to the IED (clients 1–5). This is the number of clients that can be connected simultaneously to the WAGO IEC 61850 server.
Allowed IPs	Allowed IPs for clients 1–5 Default is 0.0.0.0. If the IP address = 0.0.0.0, the validity of the IP address is not checked. If more than one client connection has been selected, additional IPs must be configured for the other clients. As soon as an IP address is parameterized with 0.0.0.0, testing for all connected clients is disabled.
Time synchronization	SNTP (default), NTP, 750-640/DCF77, 750-640/WWBV SNTP (default): SNTP time synchronization. In addition to activation, functionality must be parameterized in the device's Web-based management. Currently, the SNTP time telegram does not use milliseconds, i.e., accuracy is measured in one second increments. NTP: NTP time synchronization. NTP time synchronization requires activation of SNTP time synchronization via EthernetSettings or activation of the device's web-based management. Time zone must be parameterized with 00:00 and DST must be disabled. The time zone and DST can be set using the IEC-61850 Configurator. 750-640/DCF77: DCF77 time synchronization. The I/O module 750-640 (RTC module) with the corresponding DCF77 receiver must be used. 750-640/WWBV: GPS time synchronization. The I/O module 750-640 (RTC module) with the corresponding GPS receiver must be used.
Time zone DLS Mode	Only active if SNTP or NTP is selected during time synchronization. Time Zone: Time zone is the local time offset in relation to UTC, default = 0 hours. DLS Mode: “Daylight Savings Mode” is the local time offset for summer, default = 0 0: disabled 1: for summer, April through September.

Table 13: “Configuration” tab parameterization options

Setting Options	Description
Task1 (Com.)	Task for IEC 61850-MMS communication and (unbuffered) reporting.
Task2 (Main)	Task to test the trigger conditions in the configured dataset and, if necessary, to send GOOSE telegrams.
Keep Alive	If this function is enabled, communication nodes exchange short data telegrams cyclically to monitor the connection. Interruptions in the connection can be detected faster.

The server (IEC 61850 IED) consists of at least one “LogicalDevice” (LD). One LD consists of at least one LLN0 object and one LPHD1 object.

An LD can have any number of additional LNC objects. The number of LNCs is limited by the device memory. LDs are limited to one.

In the following figure, the LD has an additional LNC instance of the “GGIO” type.

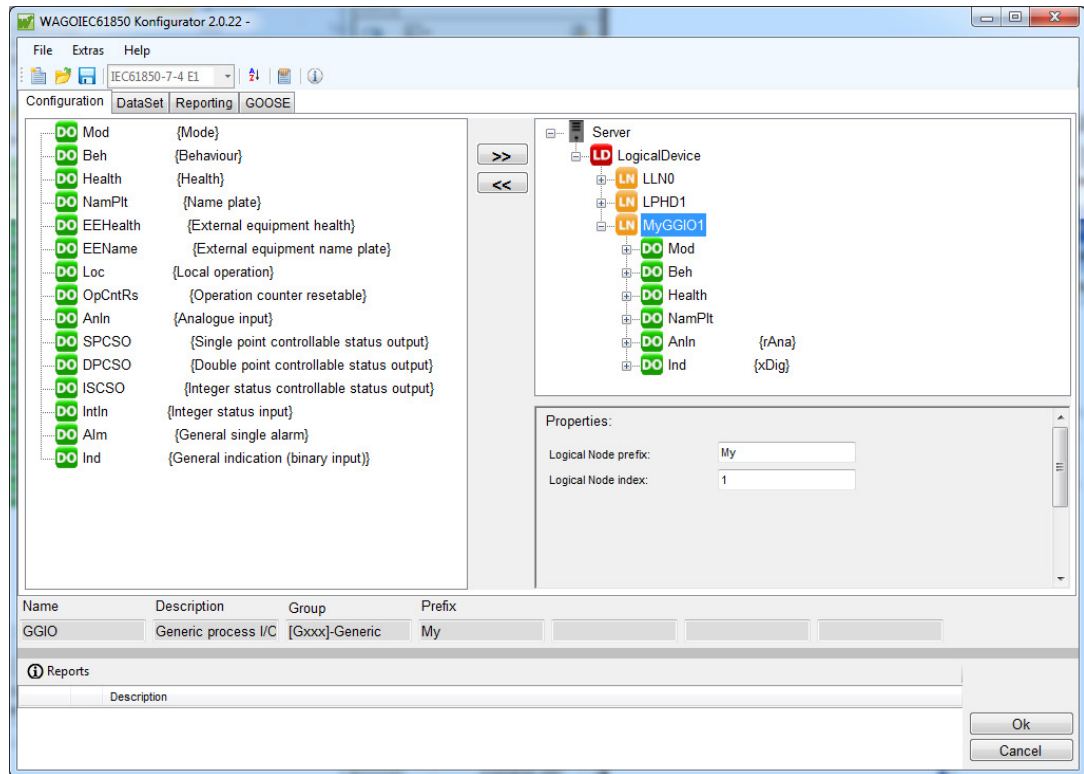


Figure 31: IED configuration – IED expanded

You have the following option to parameterize an LD:

Table 14: “LogicalDevice” parameterization option

Setting Options	Description
Device name	Name assignment for the “LogicalDevice” (maximum 64 characters)

When selecting an LNC instance, all CDC types (*mandatory* and *optional*) are displayed on the left-hand side of the window. A prefix for the name of the LN object and a trailing index can be assigned in the “Properties” area.

Now you can create additional CDCs for a given LNC instance (“GGIO1” in this example).

In the following figure, the LNC instance has the additional CDC instances “AnIn” and “Ind.”

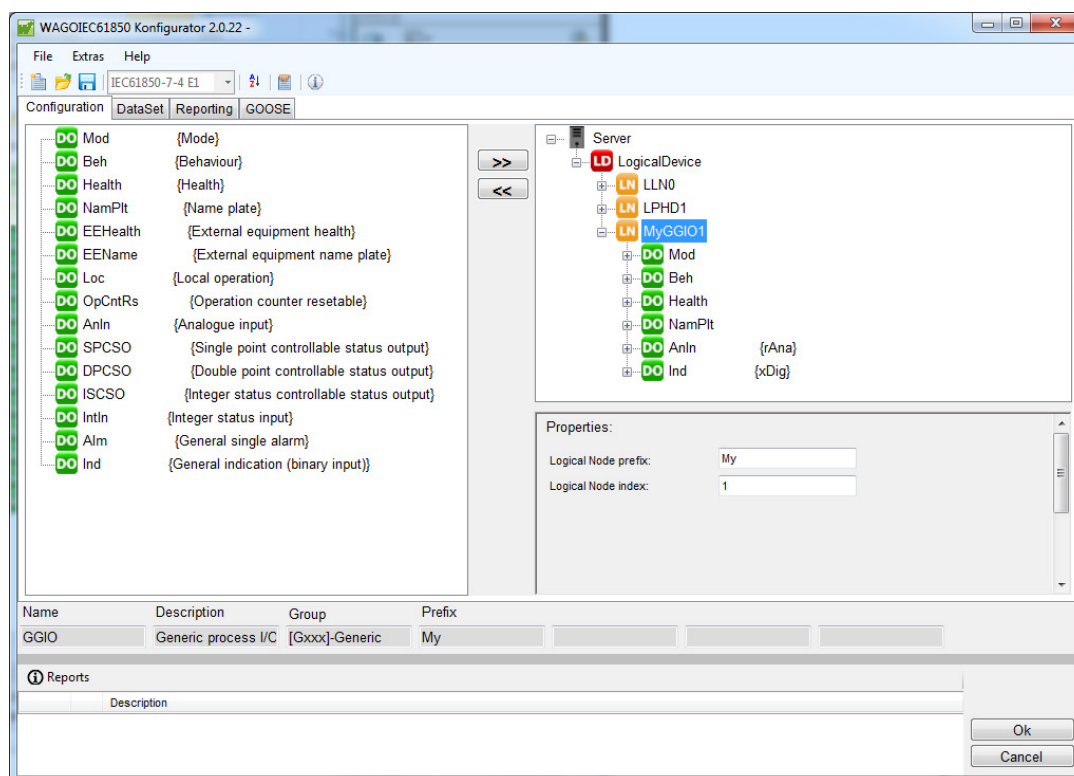


Figure 32: IED configuration – LNC instance expanded

In the Configurator, you can easily link an IEC 61850 attribute with WAGO-I/O-PRO variables. Select the CDC instance attribute to be linked and enter the required WAGO-I/O-PRO variable name under “value:” in the Properties area.

You can store an initial value in the input field to the right of the variable name. The “Autom. declare” setting means that the variable from the IEC 61850 Configurator is automatically declared in the global variable area of WAGO-I/O-PRO.

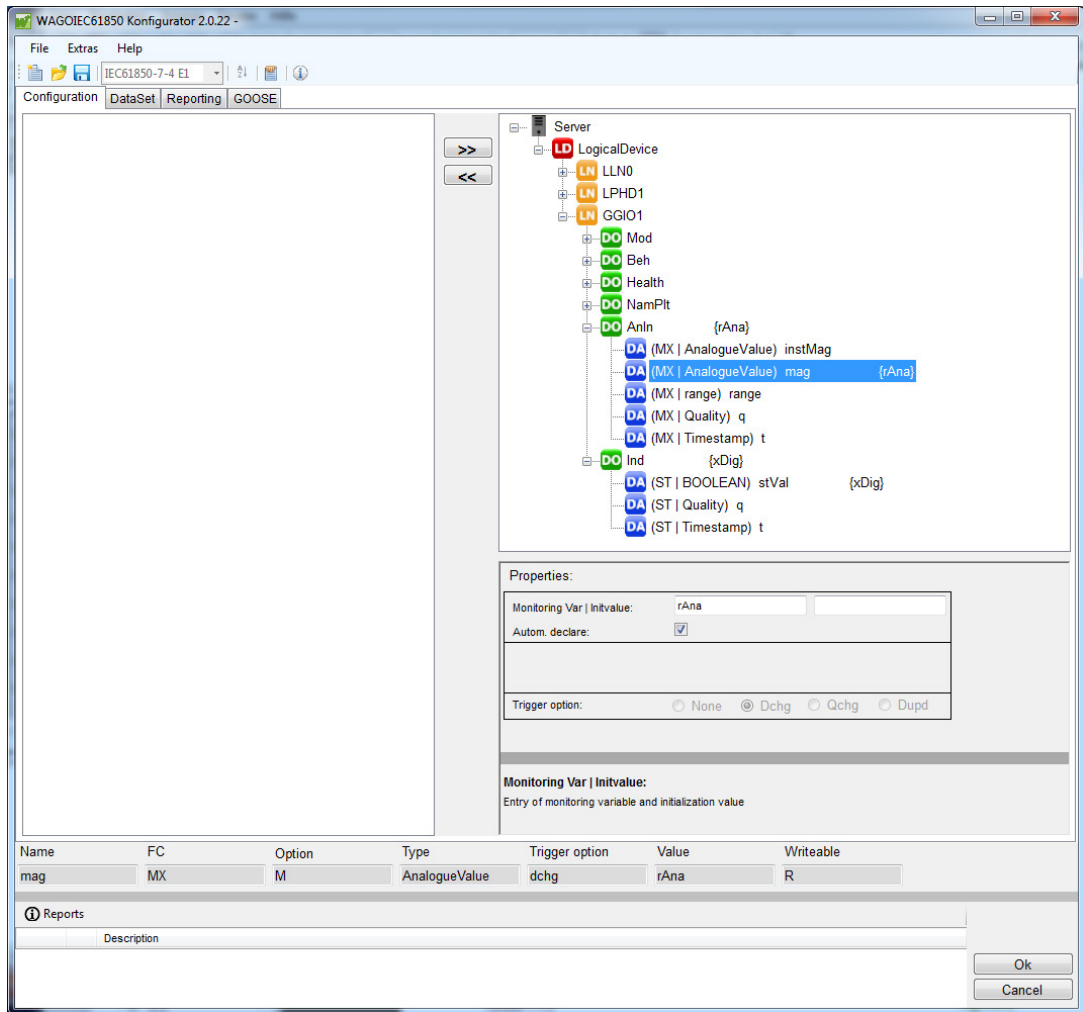


Figure 33: IED configuration – monitoring direction WAGO-I/O-PRO variable

If an attribute has RW access, another variable is visible in the control direction in addition to the variables in the monitoring direction.

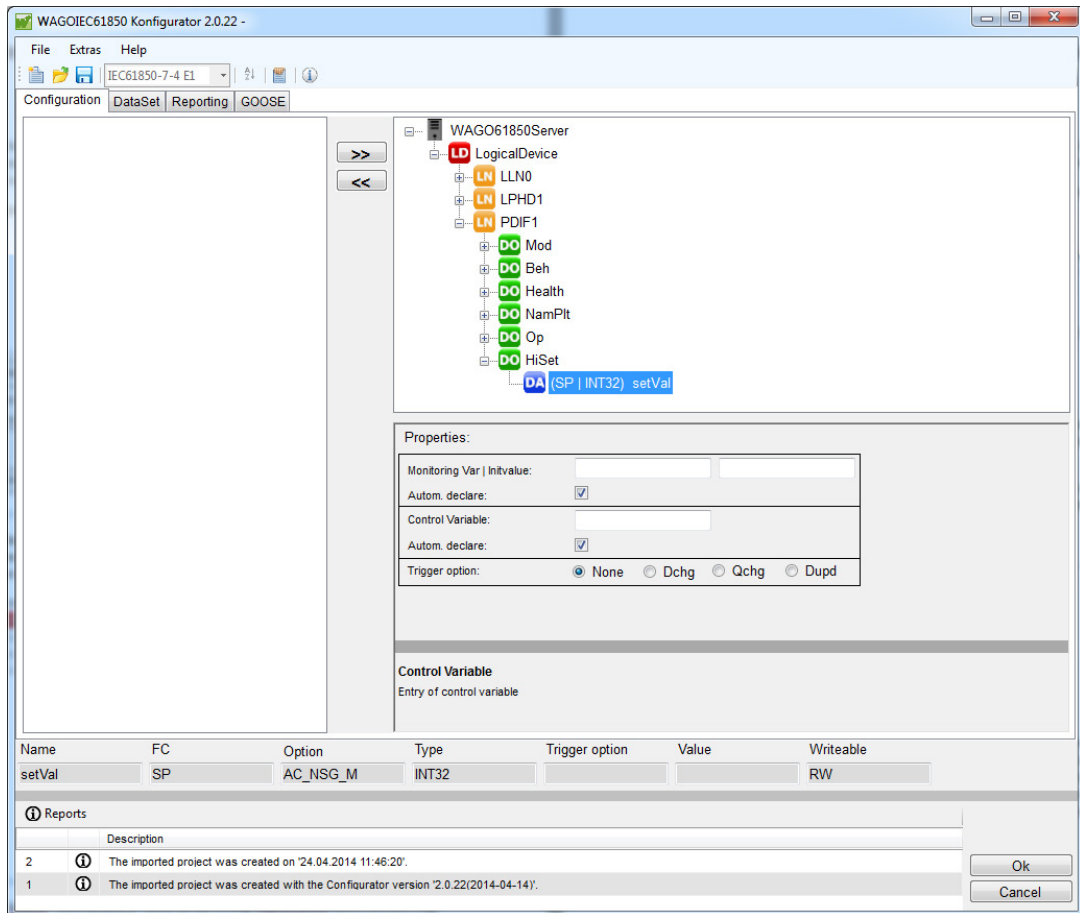


Figure 34: IED configuration – control direction WAGO-I/O-PRO variable

Note



Disable “Autom. declare” option for direct connections!

If you establish a direct connection from an I/O module channel to an IEC 61850 attribute by using the same variable name, disable the “Autom. declare” option. Should this option remain active, the variable will be declared by the Internal Data Bus Configurator and during PLC program translation. You will then receive the error message: “The same value has been declared twice.”

If the WAGO-I/O-PRO development environment is connected to the controller online, the processing time of IEC 61850 tasks for communication and data acquisition can be observed on the root node. The default settings can be optimized for task priority and cycle time.

Note



Display task cycle times graphically!

Add the “SysLibTask.lib” library to the WAGO-I/O-PRO project to display task cycle times graphically.

4.1.3.1.1 Monitoring Direction, Reading a WAGO-I/O-PRO Variable

The following illustration shows the data flow in the monitoring direction for reading a WAGO-I/O-PRO variable:

Controller → IEC 61850 client

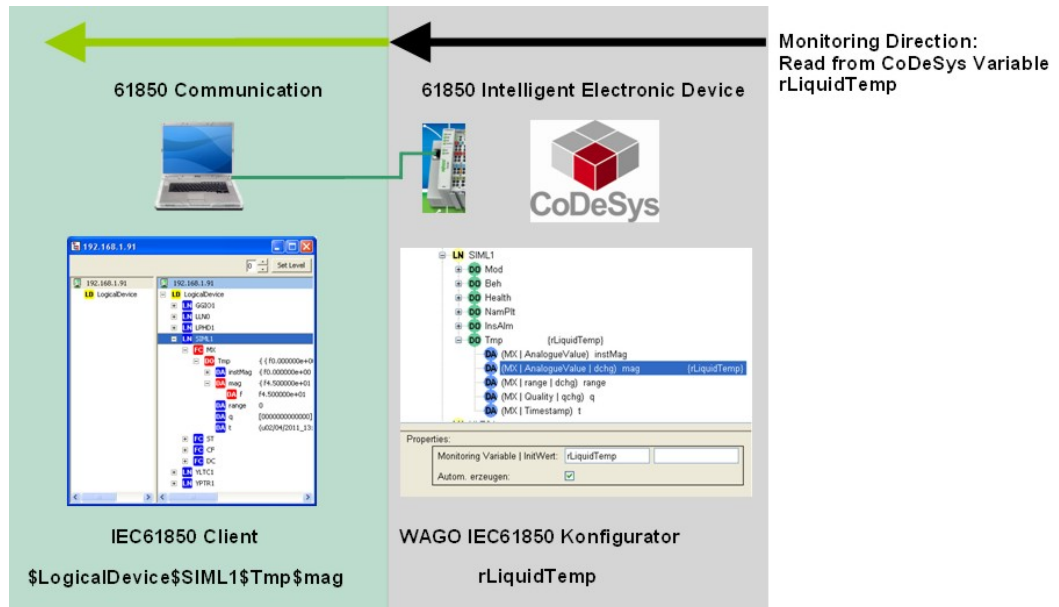


Figure 35: Monitoring direction, reading a WAGO-I/O-PRO variable

Note



Enable the “Autom. declare” option to automatically declare variables!

Make sure the “Autom. declare” option is enabled, so that the variable from the IEC 61850 Configurator is declared as a global WAGO-I/O-PRO variable. If you do not enable this option, you must manually declare the variable.

4.1.3.1.2 Monitoring Direction, Reading a Channel of the I/O Module

The following illustration shows the data flow in the monitoring direction for reading a channel of the I/O module:

I/O module → controller → IEC-61850 client

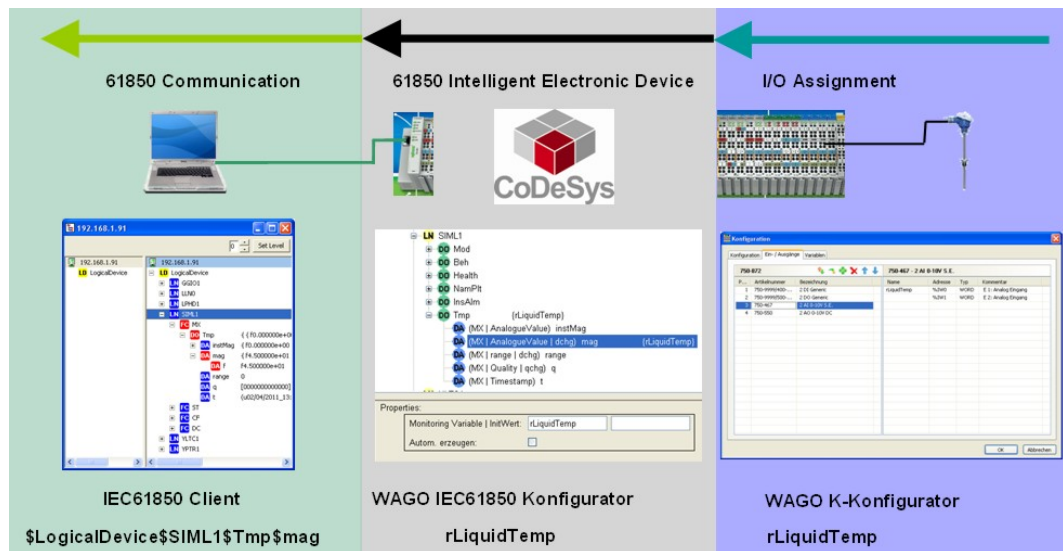


Figure 36: Monitoring direction, reading a channel of the I/O module

Note



Disable “Autom. declare” option for direct connections!

If you establish a direct connection from an I/O module channel to an IEC 61850 attribute by using the same variable name, disable the “Autom. declare” option. Should this option remain active, the variable will be declared by the Internal Data Bus Configurator and during PLC program translation. You will then receive the error message: “The same value has been declared twice.”

4.1.3.1.3 Control Direction, Writing a WAGO-I/O-PRO Variable

The following illustration shows the data flow in the control direction for writing a WAGO-I/O-PRO variable:

IEC 61850 client → controller

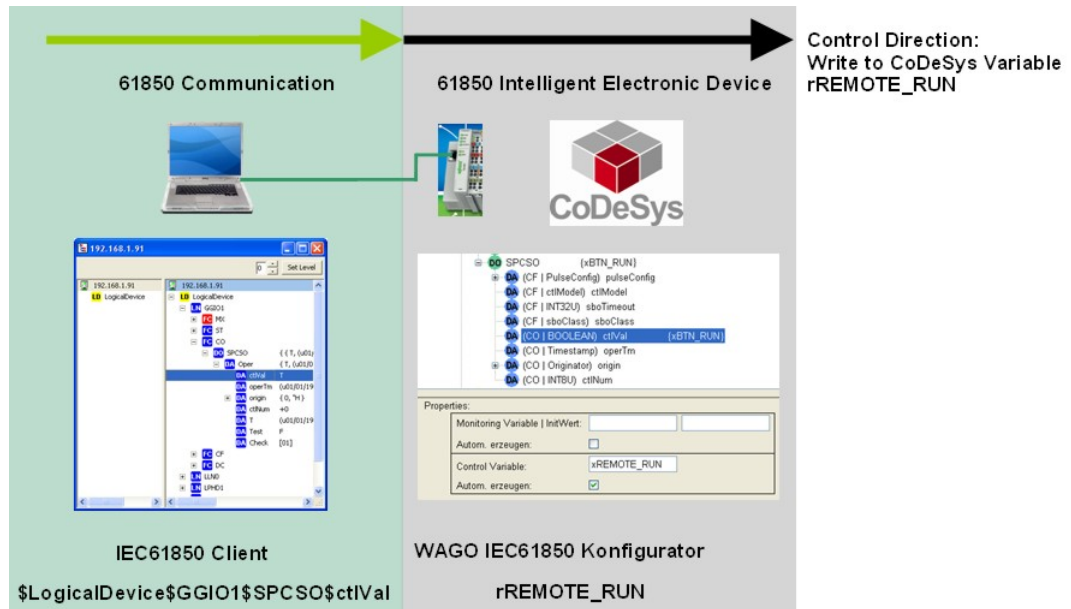


Figure 37: Control direction, writing a WAGO-I/O-PRO variable

Note



Enable the “Autom. declare” option to automatically declare variables!

Make sure the “Autom. declare” option is enabled, so that the variable from the IEC 61850 Configurator is declared as a global WAGO-I/O-PRO variable. If you do not enable this option, you must manually declare the variable.

4.1.3.1.4 Control Direction, Writing a Channel of the I/O Module

The following illustration shows the data flow in the control direction for writing an I/O module:

IEC 61850 client → controller → I/O module

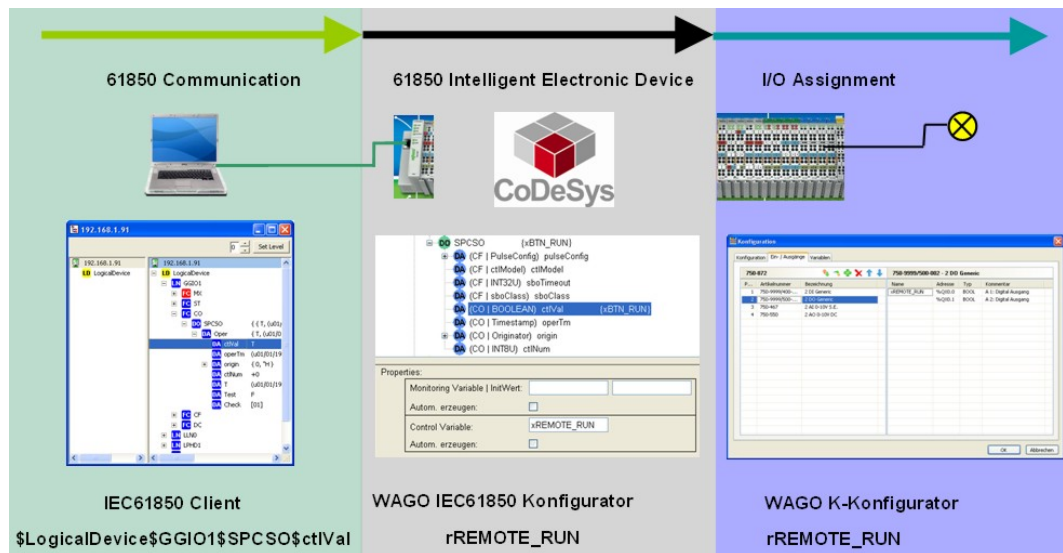


Figure 38: Control direction, writing a channel of the I/O module

Note



Disable “Autom. declare” option for direct connections!

If you establish a direct connection from an I/O module channel to an IEC 61850 attribute by using the same variable name, disable the “Autom. declare” option. Should this option remain active, the variable will be declared by the Internal Data Bus Configurator and during PLC program translation. You will then receive the error message: “The same value has been declared twice.”

4.1.3.1.5 Monitoring Direction and Control Direction, Reading and Writing a WAGO-I/O-PRO Variable

The following illustration shows the data flow in the monitoring direction for reading a WAGO-I/O-PRO variable and in the control direction for writing a WAGO-I/O-PRO variable:

Controller → IEC 61850 client and

IEC 61850 client → controller

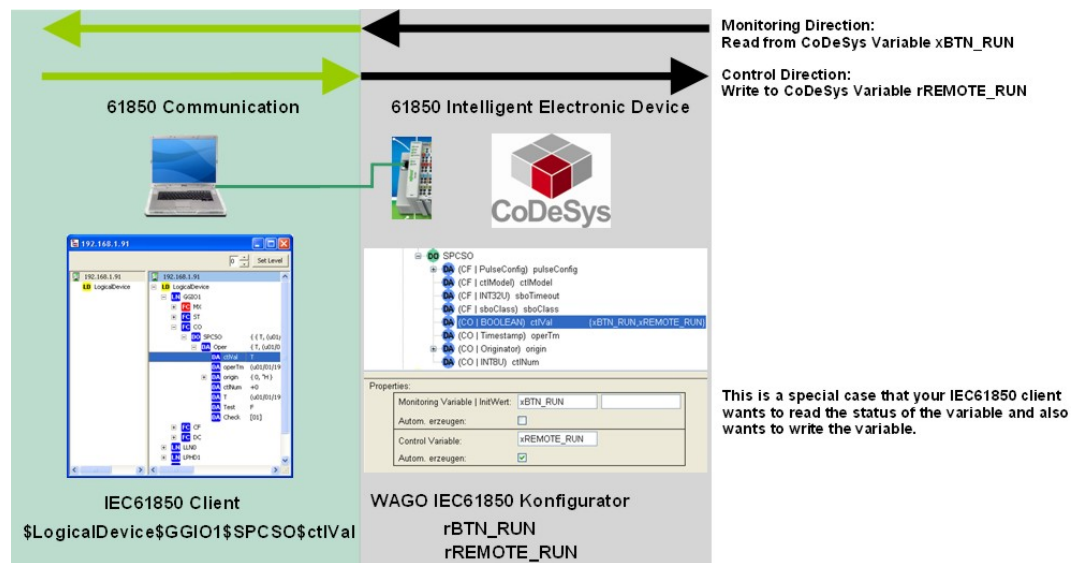


Figure 39: Monitoring direction and control direction, reading and writing a WAGO-I/O-PRO variable



Note

Enable the “Autom. declare” option to automatically declare variables!

Make sure the “Autom. declare” option is enabled, so that the variable from the IEC 61850 Configurator is declared as a global WAGO-I/O-PRO variable. If you do not enable this option, you must manually declare the variable.

4.1.3.2 “DataSet” Tab

You can create and delete datasets in the “DataSet” tab. You can also assign attributes to a dataset or delete existing assignments.

You can change the order of the entries using the “up” and “down” arrows. Select an entry in the dataset. Use the arrow keys to move the selected entry.



Note

Make sure that the entries for sender and recipient are identical!

To allow the recipient to interpret the data, the type and order of the entries from sender and recipient must be identical for GOOSE communication (see section “GOOSE’ Tab”).

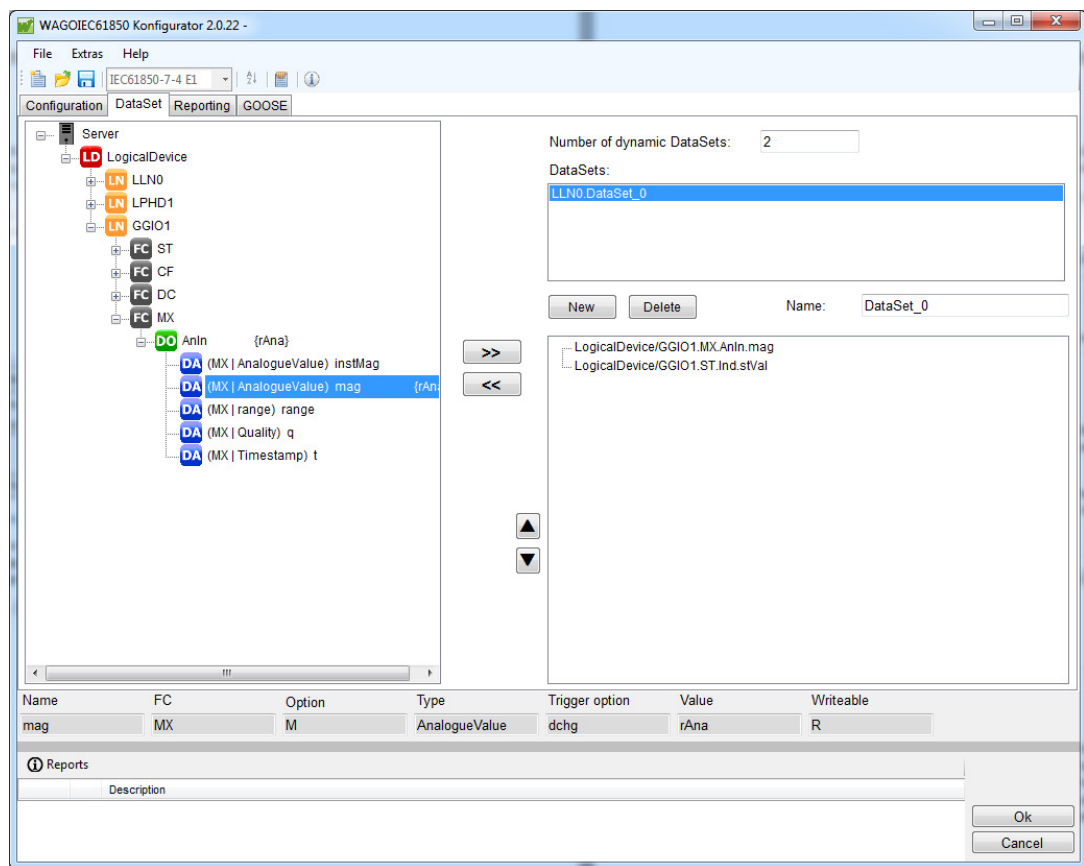


Figure 40: “DataSet” tab

Table 15: “DataSet” tab parameterization options

Setting Options	Description
[New] button	Creates a new dataset.
[Delete] button	Deletes the selected dataset.
Input field Name	Changes the name of the dataset. The name must be unique within the logical device.
[>>] button	Assigns the selected attribute to the dataset.
[<<] button	Removes the selected attribute from the dataset.
[▲] button	Moves the selected entry up.
[▼] button	Moves the selected entry down.

4.1.3.3 “Reporting” Tab

You can create and delete reports in the “Reporting” tab. You can also assign a dataset to the report and parameterize the dataset. A maximum of 20 reports can be created.

In the middle of the window, you can make general settings for the reporting configuration.

You can control the reporting behavior in the “General Options” area on the right side. When enabled, the respective field is transmitted in the message.

You can select the attributes that trigger notification of an event in the “Trigger Options” area on the right.

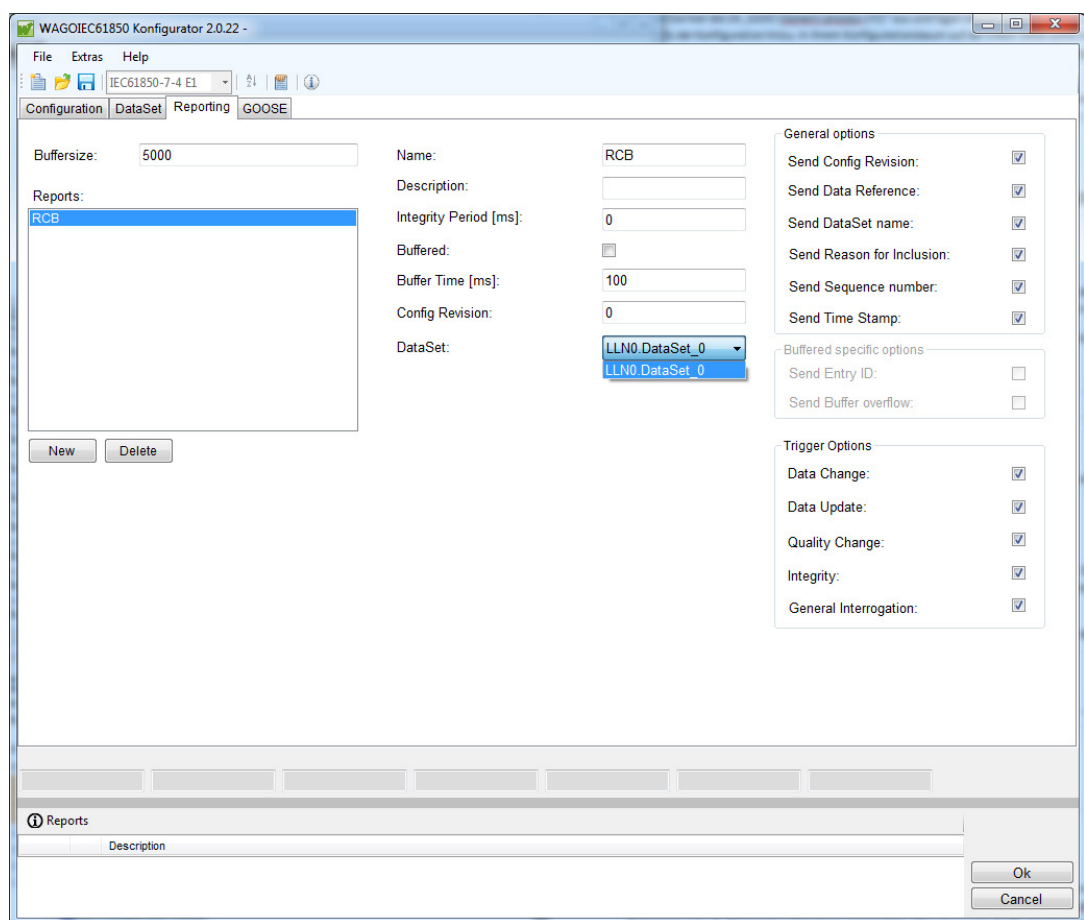


Figure 41: “Reporting” tab

Table 16: “Reporting” tab parameterization options

General Settings	
Setting Options	Description
Buffersize	Buffer size of buffered reports in bytes.
Name	Unique Report Block name within the logical node.
Description	Report Block description
Integrity Period [ms]	Gradual general query, i.e., the referenced dataset is updated after this time.

Buffered	Option, if Report Block is buffered.
	If this attribute is set to FALSE, no messages are generated and transmitted. The client can set the value to TRUE and the server begins the transmission of messages.
Buffer Time [ms]	The buffer time defines the time span that the server waits after the occurrence of an event before transmitting a message. The events occurring during this period are collected and transmitted together. Messages that come and go during this period are not recognized. If any changes should be reported continuously, then the buffer time must be 0.
Config Revision	The version identifier detects if a member is deleted from the dataset or if the order of the members has changed. Such changes mean that a value is not transmitted or the values are located at another location in the message. This is communicated to the client by a new version number. With the WAGO solution described here, identification plays no role since all datasets are fixed.
DataSet	Data set reference.
“General Options” area	
Setting Options	Description
Option “Send Config Revision”	TRUE = Attribute ConfRev is transmitted along with the notification.
Option “Send Data Reference”	TRUE = For each reported value, the complete reference is also transmitted. Example: WAGO872/GGIO.Mod.stVal
Option “Send DataSet name”	TRUE = Name of the dataset is also reported.
Option “Send Reason for Inclusion”	TRUE = For each reported value, the reason for transmission is specified.
Option “Send Sequence number”	TRUE = Messages are numbered.
Option “Send Time Stamp”	TRUE = The report has a time stamp that contains the time it was sent.
Optional area: “Buffered specific options”	
Option “Send Entry ID”	TRUE = Attribute “EntryID” is transmitted along with the notification.
Option “Send “Buffer overflow”	TRUE = Buffer overflow is reported.
“Trigger Options” area	
Setting Options	Description
Data Change	TRUE = Allow data change from the data attribute.
Data Update	TRUE = Allow data update from the data attribute.
Quality Change	TRUE = Allow quality change from the data attribute.
Integrity	TRUE = Periodic reporting of all values regardless of changes.
General Interrogation	TRUE = Dataset is sent when the client requests a general query.

4.1.3.4 “GOOSE” Tab

You can create, edit and delete GOOSE telegrams on the “GOOSE” tab.

“Generic Object Oriented Substation Events” (GOOSE) is a real-time ETHERNET protocol for fast horizontal cross-communication between “Intelligent Electronic Devices” per IEC 61850. In principle, GOOSE messages can be received by multiple recipients.

Note



750-872 Fieldbus Controller: No GOOSE communication possible!

The 750-872 Fieldbus Controller does not support GOOSE communication. This tab is hidden for this fieldbus controller.

Note



750-880/0xx-001 and 750-880/0xx-002 Fieldbus Controllers: Firmware-Release 07 or higher required!

750-880/0xx-001 and 750-880/0xx-002 Fieldbus Controllers only support GOOSE communication if firmware release 07 or higher has been installed!

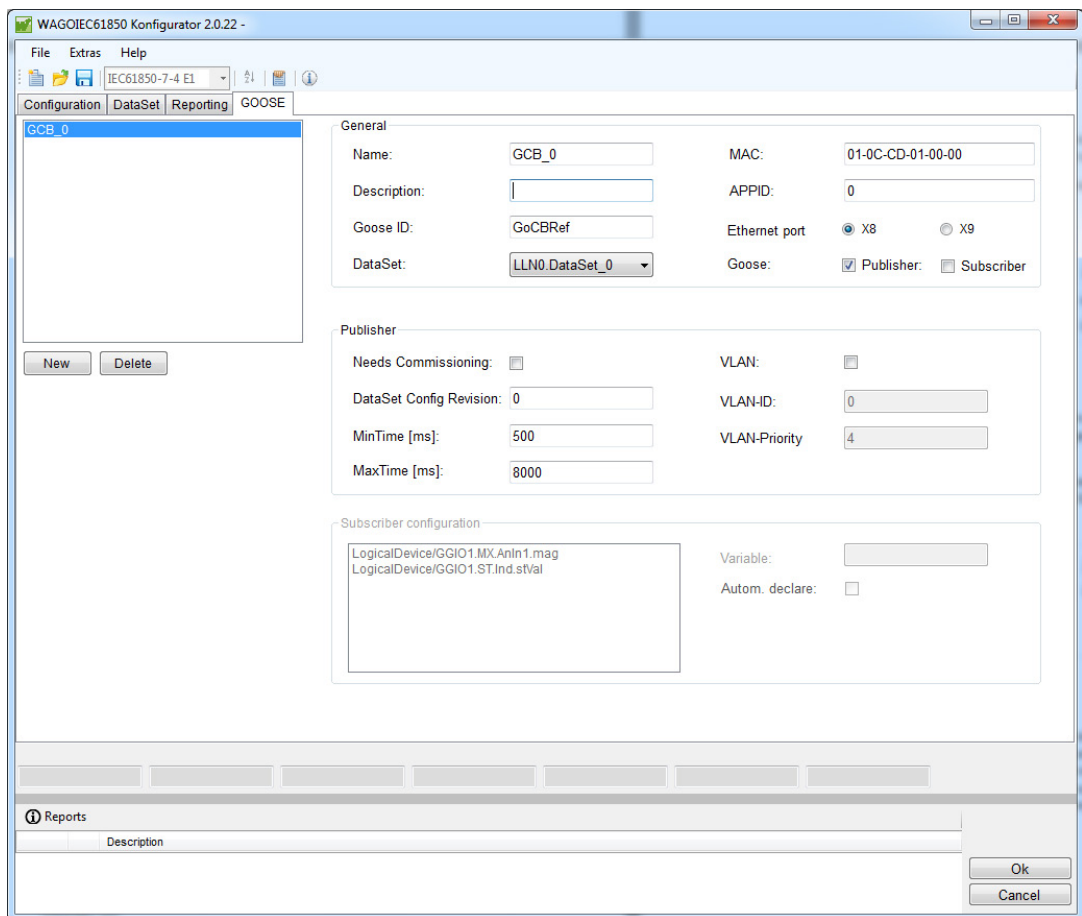


Figure 42: “GOOSE” tab

GOOSE communication is well-suited for sending several small data packets to as many recipients as possible. There is no connection monitoring via the TCP/IP layer and correct receipt of packets cannot be verified. Therefore, the delay time for data processing is much shorter. GOOSE uses a flag of the EtherType field so that data packets from the network infrastructure can be identified and prioritized.

The data is sent from a publisher and received by one or more subscribers. Data changes are transmitted immediately and the current status is cyclically.

All GOOSE messages appear in the selection list in the tab.

Note



Select GOOSE telegram!

All settings made always refer to the GOOSE telegram chosen in the selection list.

The tab offers the following parameterization options:

Table 17: “GOOSE” tab parameterization options

Setting Options	Description
[New] button	Creates another GOOSE telegram.
[Delete] button	Deletes an existing GOOSE telegram.
“General” area	
Name	Name of the GOOSE control block.
Description	Description of the GOOSE control block.
GOOSE ID	Unique character string of the GOOSE control block.
DataSet	Dataset sent or received as a GOOSE telegram.
MAC	Multicast addressing Multicast addressing sends GOOSE telegrams. Addressing allows an entire group of devices to exchange data with each other. Requirement: unique address allocation of the different device groups. Valid range of values: 01-0C-CD-01-00-00 to 01-0C-CD-01-01-FF.
APPID	Application ID Number for the system-wide unique identification of a GOOSE control block. To exchange GOOSE telegrams, this number must be identical for sender and recipient. Valid range of values: 0–4095.
ETHERNET port	X8/X9 for IPCs. Default: X8. X1/X2 for PFC200 Series Fieldbus Controllers of the and for 750-880 Fieldbus Controller . No specific interface can be selected, as these devices have only one switch.

Table 17: “GOOSE” tab parameterization options

Setting Options	Description
“Publisher” Area	
Needs Commissioning	Indicates whether the control block must be checked. The flag is not evaluated by the WAGO IEDs, but can be used by other IEDs if necessary.
DataSet Config Revision	Integer value with the version of the GOOSE control block. A subscriber can identify that the GCB has changed based on the version number.
MinTime [ms]	Minimum interval for sending GOOSE telegrams (see section “ ‘Publisher’ Area”).
MaxTime [ms]	Maximum interval for sending GOOSE telegrams (see section “ ‘Publisher’ Area”).
VLAN	“Virtual Local Area Network” Logical subnet within a physical network. Multicast messages can be passed through and filtered. The configuration is performed in managed ETHERNET switches.
VLAN-ID	A value of 0 is a non-configured VLAN in which the switch performs no filtering. This value is recommended when no logical network should be set up. Valid range of values: 0–4095.
VLAN-Priority	Messages within a managed ETHERNET switch can be forwarded depending on the priority. Valid range of values: 0–7. GOOSE default: 4.
“Subscriber Configuration” area	
Selection list	The drop-down list contains all attributes within the selected dataset. If an attribute is selected, a WAGO-I/O-PRO variable can be assigned in the input field.
Variable	The WAGO-I/O-PRO variable can be entered in this input field. The selected attribute of the incoming GOOSE telegram is then saved in this variable.
Autom. declare	The variable is automatically declared by the IEC 61850 configurator.

4.1.3.4.1 Creating a New “GOOSE Control Block” (GCB)

A requirement for creating a GCB is an existing dataset that contains objects to be sent or received. This is referenced in the GCB. It is also important that the dataset is not also referenced in a “Report Control Block” (RCB) at the same time, i.e., the dataset can only be set as a cyclic GOOSE telegram or as an MMS report. A maximum of 20 GCBs can be created.

Information



Handling multiple references to one data set!

Create two data sets with identical content to handle multiple references to one data set. Then assign each data set to one RCB and one GCB. Because less data is often desired in a GCB (e.g., no timestamp), the amount of data to be sent can be reduced.

4.1.3.4.2 “General” Area

A name for the GCB, a GOOSE ID and an APP ID can be assigned. Both IDs are used by the subscriber to uniquely identify the message. So that GOOSE telegrams can be evaluated by the subscriber, these parameters must be the same for all communication nodes. Also ensure that the dataset used is set up identically in the data structure for all communication nodes (type, number and order of the data objects).

Note



Note the size of the dataset!

The content of the dataset cannot be configured to be larger than the maximum size of an ETHERNET frame with 1526 bytes. Otherwise, no GOOSE messages are sent!

The Configurator does not check the size.

In the **MAC** input field, the MAC address range for multicast telegrams to be sent or received can be selected. The range *01-0C-CD-01-00-00* to *01-0C-01-01-FF* is provided.

4.1.3.4.3 “Publisher” Area

If a value changes in the selected dataset, the dataset is sent as a GOOSE telegram if the function is enabled. The function can be enabled by selecting the **Publisher** checkbox.

The GOOSE telegram is sent immediately after the trigger option has been enabled. When the **MinTime** has elapsed, the GOOSE telegram is transmitted again. The time interval is doubled until the **MaxTime** is reached. The GOOSE telegram is then transmitted at the **MaxTime** interval.

The values for **MinTime** and **MaxTime** can be specified in the respective input fields.

If the **VLAN** checkbox is selected, values can be entered in the **VLAN ID** and **VLAN Priority** input fields that refer to passing messages via switches.

4.1.3.4.4 “Subscriber Configuration” Area

The “Subscriber Configuration” area can only be edited when the “Subscriber” checkbox has been selected in the “General” area.

In this area, you can assign incoming GOOSE telegrams to global WAGO-I/O-*PRO* variables. Select an attribute from the list and enter a global WAGO-I/O-*PRO* variable in the input field. This variable is then written by the incoming GOOSE telegram. If the variable will be generated automatically, select the **Autom. declare** checkbox.

4.1.4 Status Bar

4.1.4.1 Status Bar of a “Logical Device Object” (LDC)

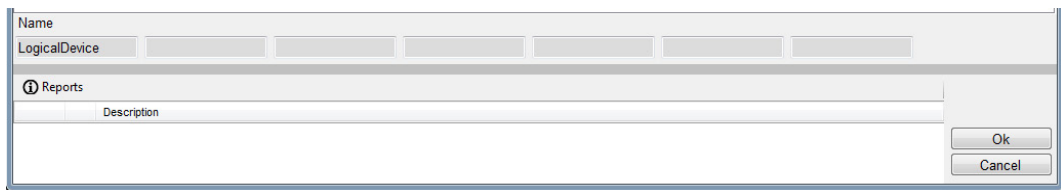


Figure 43: Status bar of a “Logical Device Object” (LDC)

Table 18: “Logical Device Object” status bar

Setting Options	Description
[OK] button	Closes the Configurator with data application and code generation.
[Cancel] button	Closes the Configurator and discards any changes.
Object information	Detailed information about the object selected. For the Server and Logical Device object type, no other detailed information. For LNC, CDC and attribute, see the following sections.

4.1.4.2 Status Bar of a “Logical Name Class Object” (LNC)

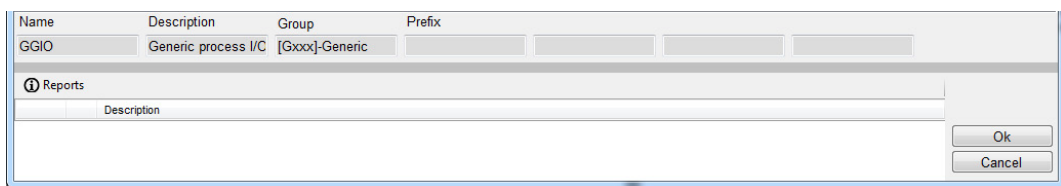


Figure 44: Status bar of a “Logical Name Class Object”

Table 19: “Logical Name Class Object” status bar

Setting Options	Description
Name	Name of selected LNC instance.
Description	Description of selected LNC instance.
Group	Related LNC grouping (e.g., Control, Generic and Protection).
Prefix	If necessary, the prefix of the LNC instance entered by the user for the name.

4.1.4.3 Status Bar of a “Common Data Class Object” (CDC)

Name	Description	Option	CDC	Instno
Anln	Analogue input	O	MV	1

Reports

Description

Ok
Cancel

Figure 45: Status bar of a “Common Data Class Object”

Table 20: “Common Data Class Object” status bar

Setting Options	Description
Name	Name of selected CDC instance.
Description	Description of selected CDC instance.
Option	Selected CDC instance option: m = mandatory o = optional
CDC	Type of selected CDC instance.
Instno	Instance number of the CDC instance. Starts with 0 for CDC, which can only be instantiated once within an LNC; otherwise, 1 to n.

4.1.4.4 Status Bar of an Attribute

Name	FC	Option	Type	Trigger option	Value	Writeable
mag	MX	M	AnalogueValue	dchg		R

Reports

Description

Ok
Cancel

Figure 46: Status bar of an attribute

Table 21: Attribute status bar

Setting Options	Description
Name	Name of the selected attribute.
FC	Functional constraint of the selected attribute.
Option	Option of the selected attribute: m = mandatory o = optional c = conditional
Type	Data type of the selected attribute.
Trigger option	Trigger options of the selected attribute. These can be: dchg = Data change dupd = Data update qchg = Quality change <empty> = No trigger option
Value	Assigned WAGO-I/O-PRO variable
Writeable	Indicates if the attribute can be read (R) and/or written (W) (viewed from the client).

4.2 “Logical Name Classes” acc. IEC 61850 / IEC 61400-25

Table 22: Support of “Logical Name Classes” acc. IEC 61850 / 61400-25

Name	Description	Support	Comment
	Automatic Control Functions		
ANCR	Neutral Current Regulator		
ARCO	Reactive Power Control		
ATCC	Automatic Tap Changer Controller	yes	
AVCO	Voltage Control		
	Control		
CALH	Alarm Handling	yes	
CCGR	Cooling Group Control	yes	
CILO	Interlocking		
CPOW	Point on Wave Switching		
CSWI	Switch Controller		
	Generic Functions		
GAPC	Generic Automatic Process Control	yes	
GGIO	Generic Process I/O	yes	
GSAL	Generic Security Application	yes	
	Interfacing and Archiving Functions		
IARC	Archiving		
IHMI	Human Machine Interface		
ITCI	Telecontrol Interface		
ITMI	Telemonitoring Interface		
	System		
LLNO	Logical Node Zero	yes	
LPHD	Physical Device Information	yes	
	Protection		
PBDF	Busbar protection		
PDCO	DC time overcurrent protection		
PDEF	Directional earth fault protection		
PDIF	Differential	yes	
PDIR	Directional		
PDIS	Distance		
PDOC	AC directional overcurrent protection		
PDOP	Directional Over Power		
PDPR	Directional power/reverse power protection		
PDOV	DC overvoltage protection		
PDUP	Directional Under Power		
PFRC	Rate of Change of Frequency	yes	
PHAR	Harmonic Restraint	yes	
PHIZ	Ground Detector	yes	
PIOC	Instantaneous Overcurrent	yes	
PLDF	Differential line protection		
PMRI	Motor Restart Inhibition	yes	
PMSS	Motor Starting Time Supervision	yes	
PNDF	Restricted earth fault protection		
POPF	Over Power Factor		
PPAM	Phase Angle Measuring		
PSCH	Protection Scheme		

Table 22: Support of “Logical Name Classes” acc. IEC 61850 / 61400-25

Name	Description	Support	Comment
PSDE	Sensitive Directional Earth Fault		
PTEF	Transient Earth Fault		
PTDF	Restricted earth fault protection		
PTOC	Time Over Current		
PTOF	Over Frequency		
PTOV	Overvoltage	yes	
PTRC	Protection Trip Conditioning		
PTTR	Thermal Overload		
PTUC	Under Current		
PTUV	Under Voltage		
PUPF	Under Power Factor		
PTUF	Under Frequency		
PVOC	Voltage Controlled Time Over Current		
PVPH	Volts per Hz		
PZSU	Zero Speed or Under Speed		
	Protection Related Functions		
RDRE	Disturbance Recorder Function		
RADR	Disturbance Recorder Channel Analog		
RBDR	Disturbance Recorder Channel Binary		
RDRS	Disturbance Record Handling		
RBRF	Breaker Failure		
RDIR	Directional Element		
RFLO	Fault Locator		
RPSB	Power Swing Detection/Blocking		
RREC	Auto Reclosing		
RSYN	Synchronism Check or Synchronizing		
	Metering and Measurement		
MDIF	Differential Measurements		
MHAI	Harmonics or Interharmonics		
MHAN	Non-Phase Related Harmonics or Interharmonics		
MMTR	Metering	yes	
MMXN	Non-Phase Related Measurement	yes	
MMXU	Measurement	yes	
MSQI	Sequence and Imbalance	yes	
MSTA	Metering Statistics	yes	
	Sensors and Monitoring		
SARC	Monitoring and Diagnostics for Arcs	yes	
SIMG	Insulation Medium Supervision (gas)	yes	
SIML	Insulation Medium Supervision (liquid)	yes	
SPDC	Monitoring and Diagnostics for Partial Discharges		
	Instrument Transformers		
TCTR	Current Transformer	yes	
TVTR	Voltage Transformer	yes	
	Wind Power Systems (IEC61400-25)		
WALM	Wind power plant alarm information	yes	
WALG	Wind turbine analog log information		
WAPC	Wind power plant active power control information	yes	
WCNV	Wind turbine converter information	yes	
WGEN	Wind turbine generator information	yes	
WMET	Wind power plant meteorological information	yes	
WNAC	Wind turbine nacelle information	yes	
WREP	Wind turbine report information		

Table 22: Support of “Logical Name Classes” acc. IEC 61850 / 61400-25

Name	Description	Support	Comment
WROT	Wind turbine rotor information	yes	
WRPC	Wind power plant reactive power control information	yes	
WSLG	Wind turbine state log information		
WTOW	Wind turbine tower information	yes	
WTRF	Wind turbine transformer information	yes	
WTRM	Wind turbine transmission information	yes	
WTUR	Wind turbine general information	yes	
WYAW	Wind turbine yawing information	yes	
	X – Switchgear Functions		
XCBR	Circuit Breaker	yes	
XSWI	Circuit Switch	yes	
	Y – Power Transformers		
YEFN	Earth Fault Neutralizer	yes	
YLTC	Tap Changer	yes	
YPSH	Power Shunt	yes	
YPTR	Power Transformer	yes	
	Further Power Systems Equipment		
ZAXN	Auxilliary Network	yes	
ZBAT	Battery	yes	
ZBSN	Bushing		
ZCAB	Power Cable		
ZCAP	Capacitor Bank	yes	
ZCON	Converter	yes	
ZGEN	Generator	yes	
ZGIL	Gas Insulated Line	yes	
ZLIN	Power Overhead Line	yes	
ZMOT	Motor	yes	
ZREA	Reactor	yes	
ZRRC	Rotating reactive component	yes	
ZSAR	Surge arrester	yes	
ZTCF	Thyristor controlled frequency converter	yes	
ZTCR	Thyristor controlled reactive converter	yes	

4.3 IEC 61850 Functionality

Table 23: Support of IEC 61850 functionalities

Functionality	Support	Comment
Logical device	yes	
Logical node	yes	
Data set	yes	Can also be defined for run-time.
Substitution		
Setting group control		
Buffered report control	yes	
Sequence number	yes	
Report time stamp	yes	
Reason for inclusion	yes	
Data set name	yes	
Data reference	yes	
Buffer overflow	yes	
Entry-ID	yes	
Buffer Time	yes	
Integrity Period	yes	
General Interrogation	yes	
Config Revision	yes	
Unbuffered report control	yes	
Sequence number	yes	
Report time stamp	yes	
Reason for inclusion	yes	
Data set name	yes	
Data reference	yes	
Buffer Time	yes	
Integrity Period	yes	
General Interrogation	yes	
Config Revision	yes	
Logging		
Log control		
Log		
Control	yes	
GOOSE		750-880/0xx-001 (firmware release 07), 750-880/0xx-002 (firmware release 07), 750-8202/025-001, 750-8202/025-002, 750-8206/025-001, 758-874/000-130, 758-874/000-131.
GSSE		
Time	yes	
File Transfer		
Maximum number of simultaneous client connections	Up to five, as defined by compile time	
Maximum MMS PDU size	16 Kbytes	
Time synchronization	(S)NTP, WAGO 750-640 (DCF, GPS)	
SCL File support	Export/import into WAGO Configuration Tool	

4.3.1 Project Planning and Design

Note



Check selected controller for individual application purpose!

In *WAGO-I/O-PRO*, you can check whether a selected controller is suitable for your individual application. The respective target must be selected and the source text generated with a configuration according to the application. After the source text is compiled, memory usage of the device is displayed. This depends on the configured LNCs and, if necessary, CDCs and attributes. The number of configured clients also has an impact because object instances are created for each client.

List of Figures

Figure 1: Example of Transformer Monitoring Application with Connected IEC 61850 Client.....	14
Figure 2: “PLC Configuration” workspace.....	16
Figure 3: IED configuration, IED expanded by one LNC instance of the “GGIO” type.....	17
Figure 4: IED configuration, generate additional CDC instance.....	18
Figure 5: IED configuration, monitoring direction: linking IEC 61850 attributes to WAGO-I/O-PRO variable.....	19
Figure 6: Dataset configuration.....	20
Figure 7: Reporting configuration.....	21
Figure 8: Status bar buttons to apply or discard changes.....	22
Figure 9: Generated and compiled IEC 61850 solution in WAGO-I/O-PRO.....	22
Figure 10: “IEDScout” with object structure and dataset connected to WAGO controller.....	24
Figure 11: “Configuration” tab.....	27
Figure 12: Generic logical nodes.....	28
Figure 13: Variable assignment for the analog input.....	28
Figure 14: “DataSet” tab.....	30
Figure 15: Configure dataset.....	31
Figure 16: “GOOSE” tab.....	32
Figure 17: General GOOSE settings for a PFC200.....	33
Figure 18: Subscriber configuration.....	34
Figure 19: GOOSE telegram in Wireshark.....	35
Figure 20: Global variables.....	36
Figure 21: Publisher global variables.....	36
Figure 22: “Write Variable rAnValue” window.....	37
Figure 23: Writting a variable value.....	37
Figure 24: Global variables of the subscriber.....	37
Figure 25: IEC 61850 Configurator overview.....	38
Figure 26: Configurator “File” menu.....	39
Figure 27: Configurator “Tools” menu.....	40
Figure 28: “Options” dialog.....	40
Figure 29: Configurator “Help” menu.....	41
Figure 30: “Configuration” tab.....	42
Figure 31: IED configuration – IED expanded.....	44
Figure 32: IED configuration – LNC instance expanded.....	45
Figure 33: IED configuration – monitoring direction WAGO-I/O-PRO variable.....	46
Figure 34: IED configuration – control direction WAGO-I/O-PRO variable.....	47
Figure 35: Monitoring direction, reading a WAGO-I/O-PRO variable.....	48
Figure 36: Monitoring direction, reading a channel of the I/O module.....	49
Figure 37: Control direction, writing a WAGO-I/O-PRO variable.....	50
Figure 38: Control direction, writing a channel of the I/O module.....	51
Figure 39: Monitoring direction and control direction, reading and writing a WAGO-I/O-PRO variable.....	52
Figure 40: “DataSet” tab.....	53
Figure 41: “Reporting” tab.....	54
Figure 42: “GOOSE” tab.....	56
Figure 43: Status bar of a “Logical Device Object” (LDC).....	61

Figure 44: Status bar of a “Logical Name Class Object”	61
Figure 45: Status bar of a “Common Data Class Object”	62
Figure 46: Status bar of an attribute	62

List of Tables

Table 1: Number Notation.....	8
Table 2: Font Conventions	8
Table 3: Abbreviations	13
Table 4: Hardware Requirements.....	15
Table 5: Key for “Configuration tab” figure	27
Table 6: Key for “DataSet tab” figure.....	30
Table 7: Key for “GOOSE tab” figure	32
Table 8: Key for IEC 61850 Configurator overview	39
Table 9: Configurator “File” menu	39
Table 10: Configurator “Tools” menu.....	40
Table 11: Key for “Options dialog” figure.....	40
Table 12: Configurator “Help” menu.....	41
Table 13: “Configuration” tab parameterization options	43
Table 14: “LogicalDevice” parameterization option.....	44
Table 15: “DataSet” tab parameterization options	53
Table 16: “Reporting” tab parameterization options.....	54
Table 17: “GOOSE” tab parameterization options	57
Table 18: “Logical Device Object” status bar.....	61
Table 19: “Logical Name Class Object” status bar	61
Table 20: “Common Data Class Object” status bar	62
Table 21: Attribute status bar	62
Table 22: Support of “Logical Name Classes” acc. IEC 61850 / 61400-25	63
Table 23: Support of IEC 61850 functionalities	66

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: +49/5 71/8 87 – 0
Fax: +49/5 71/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>



This Page Left Intentionally Blank



WAGO-I/O-SYSTEM 750 **DNP3 Solution for programmable Controls of** **Telecontrol Technology** **759-911**

Version 1.0.0



© 2015 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

1	Notes about this Documentation.....	5
1.1	Copyright.....	5
1.2	Symbols.....	6
1.3	Number Notation.....	8
1.4	Font Conventions.....	8
2	Important Notes.....	9
2.1	Legal Bases.....	9
2.1.1	Subject to Changes.....	9
2.1.2	Personnel Qualifications.....	9
2.1.3	Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions.....	9
2.1.4	Technical Condition of Specified Devices.....	10
2.2	Safety Advice (Precautions).....	11
3	Application Description.....	13
3.1	Abbreviations.....	13
3.2	General Information about the DNP3 Standard.....	14
3.3	Task Definition.....	14
3.4	Hardware Requirements.....	15
3.5	Software Requirements.....	15
3.6	Create an Example Project in DNP3 and Open the DNP3 Configurator.....	16
3.7	Create DNP3 Configuration.....	18
3.8	Open Program, Assign Values and Start Program.....	21
3.9	Simulation of a Telecontrol Station.....	22
3.10	Set the General Parameters.....	22
4	DNP3 Configurator.....	25
4.1	Menu Bar.....	26
4.1.1	“File” Menu.....	26
4.1.2	“Configuration” Menu.....	27
4.1.3	“Help” Menu.....	27
4.2	Workspace.....	28
4.2.1	Global Parameters.....	29
4.2.2	“Link Layer” Parameters.....	31
4.2.2.1	“Connector Ethernet” Object.....	31
4.2.2.2	“Connector Serial” Object.....	33
4.2.3	“Application Layer” Parameters.....	34
4.2.3.1	“Group Binary Input” Group Object.....	34
4.2.3.2	“Group Binary Output” Group Object.....	37
4.2.3.3	“Group Counter” Group Object.....	40
4.2.3.4	“Group Analog Input” Group Object.....	43
4.2.3.5	“Group Analog Output” Group Object.....	48
4.3	Status Bar.....	53
4.4	Diagnostics Messages via Dialog.....	55
5	Appendix.....	57
6	Glossary.....	59

List of Figures 63
List of Tables..... 64

1 Notes about this Documentation

1.1 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.2 Symbols

 **DANGER**

Personal Injury!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **DANGER**

Personal Injury Caused by Electric Current!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **WARNING**

Personal Injury!

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

 **CAUTION**

Personal Injury!

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE

Damage to Property Caused by Electrostatic Discharge (ESD)!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note

Important Note!

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.

Information

**Additional Information:**

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.3 Number Notation

Table 1: Number Notation

Number Code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.4 Font Conventions

Table 2: Font Conventions

Font Type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Program Files\WAGO Software</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualifications

All sequences implemented on Series 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

All changes to the coupler or controller should always be carried out by qualified personnel with sufficient skills in PLC programming.

Basic knowledge of the DNP3 standard is also required.

2.1.3 Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions

Fieldbus couplers, fieldbus controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to actuators or higher-level control systems. Using programmable controllers, the signals can also be (pre-) processed.

The devices have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the devices in wet and dusty environments is prohibited.

Operating the WAGO-I/O-SYSTEM 750 devices in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section “Device Description” > “Standards and Guidelines” in the manual for the used fieldbus coupler/controller.

Appropriate housing (per 94/9/EG) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

2.1.4 Technical Condition of Specified Devices

The devices to be supplied ex works are equipped with hardware and software configurations, which meet the individual application requirements. WAGO Kontakttechnik GmbH & Co. KG will be exempted from any liability in case of changes in hardware or software as well as to non-compliant usage of devices.

Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

2.2 Safety Advice (Precautions)

For installing and operating purposes of the relevant device to your system the following safety precautions shall be observed:



DANGER

Do not work on devices while energized!

All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.

DANGER

Install the device only in appropriate housings, cabinets or in electrical operation rooms!

The WAGO-I/O-SYSTEM 750 and its components are an open system. As such, install the system and its components exclusively in appropriate housings, cabinets or in electrical operation rooms. Allow access to such equipment and fixtures to authorized, qualified staff only by means of specific keys or tools.

NOTICE

Replace defective or damaged devices!

Replace defective or damaged device/module (e.g., in the event of deformed contacts), since the long-term functionality of device/module involved can no longer be ensured.

NOTICE

Protect the components against materials having seeping and insulating properties!

The components are not resistant to materials having seeping and insulating properties such as: aerosols, silicones and triglycerides (found in some hand creams). If you cannot exclude that such materials will appear in the component environment, then install the components in an enclosure being resistant to the above-mentioned materials. Clean tools and materials are imperative for handling devices/modules.

NOTICE

Clean only with permitted materials!

Clean soiled contacts using oil-free compressed air or with ethyl alcohol and leather cloths.

NOTICE**Do not use any contact spray!**

Do not use any contact spray. The spray may impair contact area functionality in connection with contamination.

NOTICE**Do not reverse the polarity of connection lines!**

Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

NOTICE**Avoid electrostatic discharge!**

The devices are equipped with electronic components that may be destroyed by electrostatic discharge when touched. Please observe the safety precautions against electrostatic discharge per DIN EN 61340-5-1/-3. When handling the devices, please ensure that environmental factors (personnel, work space and packaging) are properly grounded.

3 Application Description

3.1 Abbreviations

Table 3: Abbreviations

Abbreviation	Explanation	Description
ASDU	Application Service Data Unit	Summary of the application services of a data unit. Also specifically designates the common address for identifying all information objects for a specific substation.
CRC	Cyclic Redundancy Check	Procedure for checking the consistency of data
CSV	Comma-separated Values	Text file format for saving measured values and parameters
DNP3	The Third Generation of Distributed Network Protocol	Communications protocol for telecontrol applications
FT#	Frame Type #	Frame format for data telegrams
IEC	International Electrotechnical Commission	International standardization organization for electrical and electronics systems
IED	Intelligent Electronic Device	An IED is an intelligent field device based on DNP3.
IEEE	Institute of Electrical and Electronics Engineers	Global professional association of engineers which also make up standardization committees
IIN	Internal Indications	Internal field consisting of bits with status information within a DNP3 user data telegram
IP	Internet Protocol	Non-connection-based network protocol at the network layer level
RTU	Remote Terminal Unit	Designation for a telecontrol substation
SCADA	Supervisory Control and Data Acquisition	Monitoring and control of technical procedures and processes using a computer system
TCP	Transmission Control Protocol	Connection-oriented network protocol at the transport layer level
UDP	User Datagram Protocol	Non-connection-based network protocol at the transport layer level

3.2 General Information about the DNP3 Standard



Information

Observe the functions supported in line with DNP3!

A list of supported functions for DNP3 is given in the Download area at www.wago.com.

The DNP3 standard also features the following:

- The name key for the data point transmission format according to DNP3 consists of group and variation numbers. A sample nomenclature is **10.02**. Here, the **10** represents the group number and the **02** stands for the variation number.
- Telegrams in the control direction can include more than one command/setpoint. Commands can also activate complex patterns and sequences with repetitions in each telegram.
- The current statuses of command or setpoint outputs can be queried or read back.
- Each telegram with user data transmitted from the substation to the master station contains a 16-bit string, which contains status information, for example events at the substation.
- DNP3 also defines a “frozen” status for measured values and counters. This enables many or all values to be frozen at a point in time and to then transfer these values to the master station. If the “frozen” command is issued in a broadcast, values can simultaneously be frozen and then transmitted at several substations.
- Changes in values at the substation are handled using events. Measured values, counters or messages can be allocated to one of the event categories (1, 2 or 3) or to category 0 (no events). The user can specify the definition of event categories 1, 2 or 3.

3.3 Task Definition

This Quickstart Guide explains how to set up and start a telecontrol substation (server) based on DNP3 with WAGO controllers.

The DNP3 Configurator (v1.00.00 or higher) integrated into WAGO-I/O-PRO is used as the configuration program. To check operation, a connection must be established between the controller and PC.

3.4 Hardware Requirements

The following WAGO hardware components can be used (hereafter referred to as “target system”):

Table 4: Hardware Requirements

Hardware Components	Item Number	Release Version	Serial Interface	Specific Requirements for DNP3
Programmable Fieldbus controller for telecontrol applications	750-872	Release version (04) or higher	COM 0	Link connections: max. 4 Event buffer: < 8000 Information objects: < 128
Programmable Fieldbus controller for telecontrol applications	750-872/020-000	Release version (04) or higher		Link connections: max. 4 Event buffer: < 8000 Information objects: < 128
Programmable ETHERNET controller for telecontrol applications	750-880/025-001	Release version (04) or higher		Link connections: max. 4 Event buffer: < 8000 Information objects: < 255
Programmable ETHERNET controller for telecontrol applications	750-880/025-002	Release version (04) or higher		Link connections: max. 4 Event buffer: < 8000 Information objects: < 255
Programmable ETHERNET controller for telecontrol applications	750-880/040-001	Release version (04) or higher		Link connections: max. 4 Event buffer: < 8000 Information objects: < 255
I/O-IPC-C6 Linux [®] Telecontrol	758-874/000-130	Release version (08) or higher	COM 1	Link connections: max. 4 Event buffer: < 16000 Information objects: < 1024
I/O-IPC-C6 Linux [®] TeleControl PDP Master	758-874/000-131	Release version (08) or higher	COM 1	Link connections: max. 4 Event buffer: < 16000 Information objects: < 1024
PFC200 CS 2ETH RS TeleControl/T	750-8202/025-001	Release version (01) or higher	COM 0	Link connections: max. 4 Event buffer: < 16000 Information objects: < 1024
PFC200 CS 2ETH RS TeleControl ECO/T	750-8202/025-002	Release version (01) or higher	COM 0	Link connections: max. 4 Event buffer: < 16000 Information objects: < 1024
PFC200 CS 2ETH RS CAN DPS TeleControl/T	750-8206/025-001	Release version (03) or higher	COM 0	Link connections: max. 4 Event buffer: < 16000 Information objects: < 1024

3.5 Software Requirements

- WAGO-I/O-PRO v2.3.9.46 or higher
- Client software based on DNP3 for the master station, e.g., “DNP3 protocol (Source Forge)” software.

3.6 Create an Example Project in DNP3 and Open the DNP3 Configurator

This section describes how the DNP3 Configurator can be used to establish communication according to DNP3.

To start the project, take the following steps:

1. Launch WAGO-I/O-PRO.
2. Create a new project. Click **File** > **New**.

The “Target Settings” dialog appears.

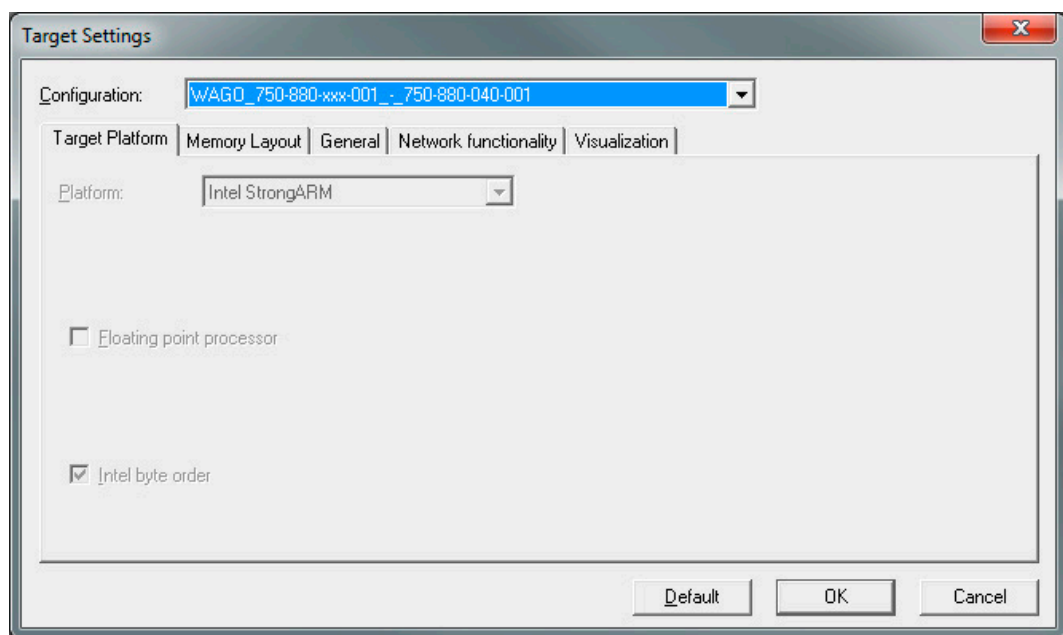


Figure 1: “Target Settings” Dialog

3. Under **Configuration**, select the hardware you are using.
4. Click on **[OK]**.

The “New POU” dialog appears.

5. Click on **[OK]**.

A new module is created.

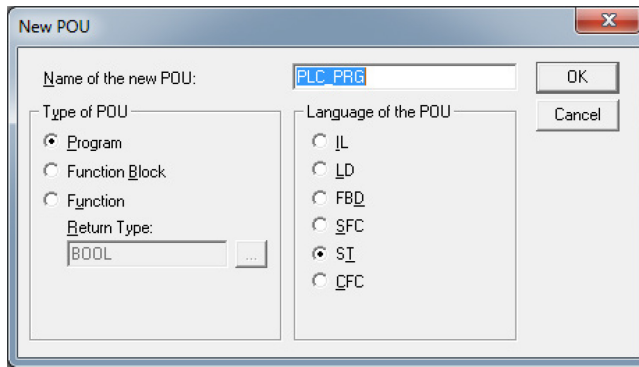


Figure 2: “New POU” dialog

6. In the program editor “PLC_PRG (PRG-ST)” enter a semicolon.

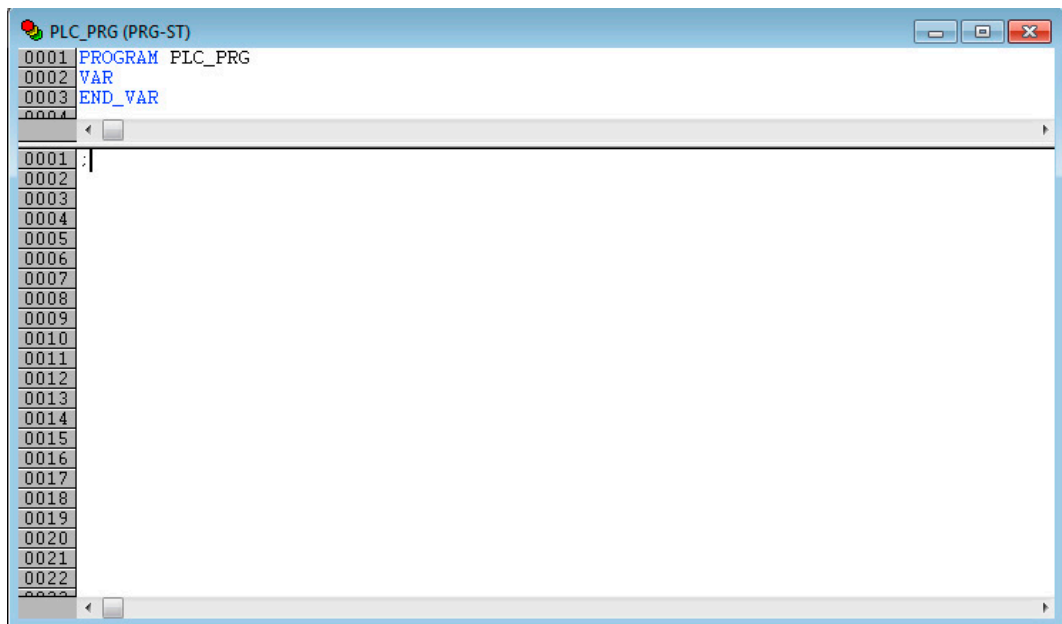


Figure 3: Program Editor “PLC_PRG (PRG-ST)”

7. Select the **Resources** tab.
 8. In the directory, select the **PLC Configuration** entry.
- The “PLC Configuration” workspace appears.
9. Select the **DNP3 Config(FIX)** entry in the tree structure.
- The “DNP3 Configurator” dialog appears.

3.7 Create DNP3 Configuration

1. Click on **Link Layer** in the “Configuration” area.
2. Click on the object **Connector Ethernet** in the “Choice” area.

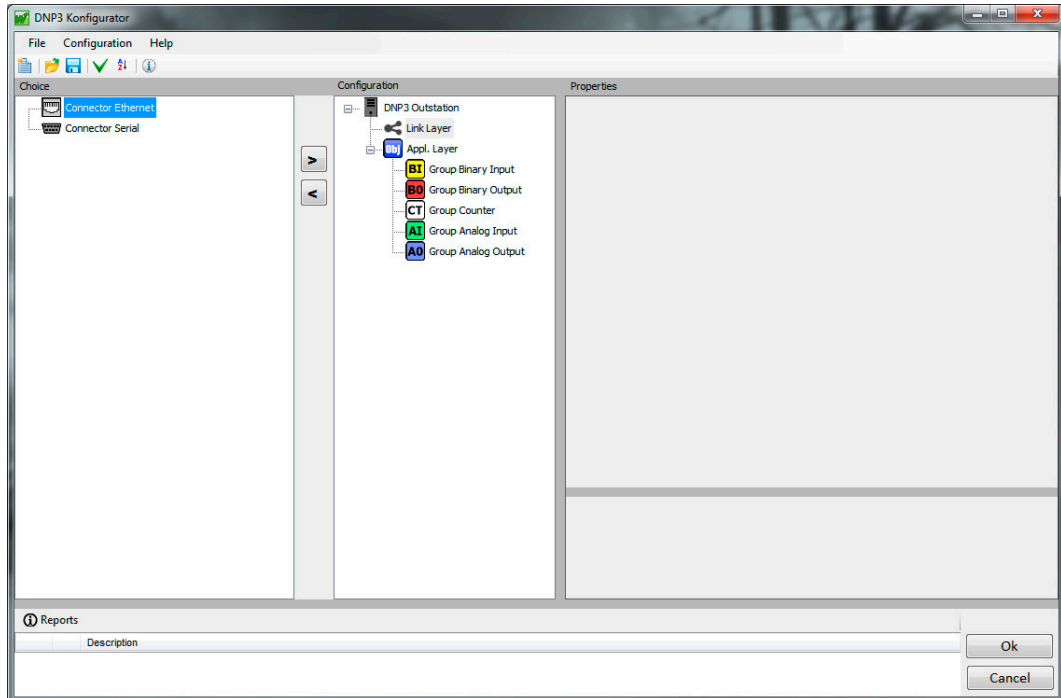


Figure 4: DNP3 Konfigurator – Add “Connector Ethernet” Object

3. Click on the button [>>] to apply the object in the “Configuration” area.
4. Click on the object **Connector Ethernet** in the “Configuration” area.
5. Make the following settings one after the other in the “Properties” area.
 - In the “Slave Addr.” input field, enter the ID number **3**.
 - In the “Master Addr.” input field, enter the ID number **4**.

Information



Disable DHCP in the Windows dialog!

Always ensure that the “DHCP” setting is disabled in the Windows dialog!

- In the “Permitted IP 1” input field, enter the IP address of your computer.

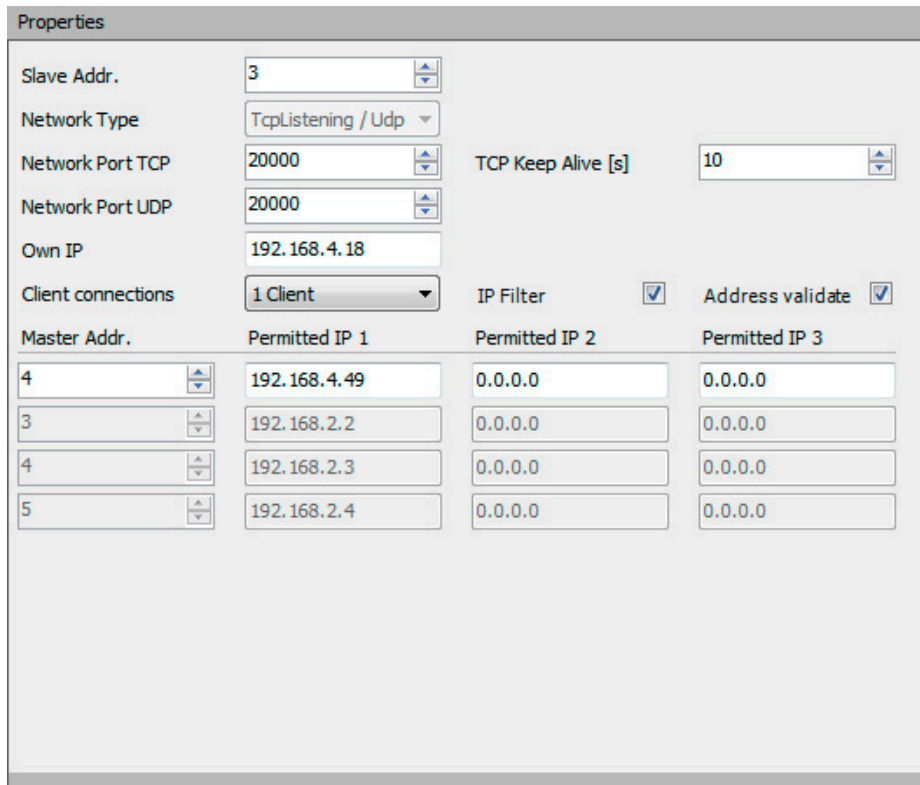


Figure 5: Required Settings for the “Connector Ethernet” Object

6. In the “Configuration” area, click on the **Group Analog Input** object.
7. Click on the information object **Analog Input** in the “Choice” area.

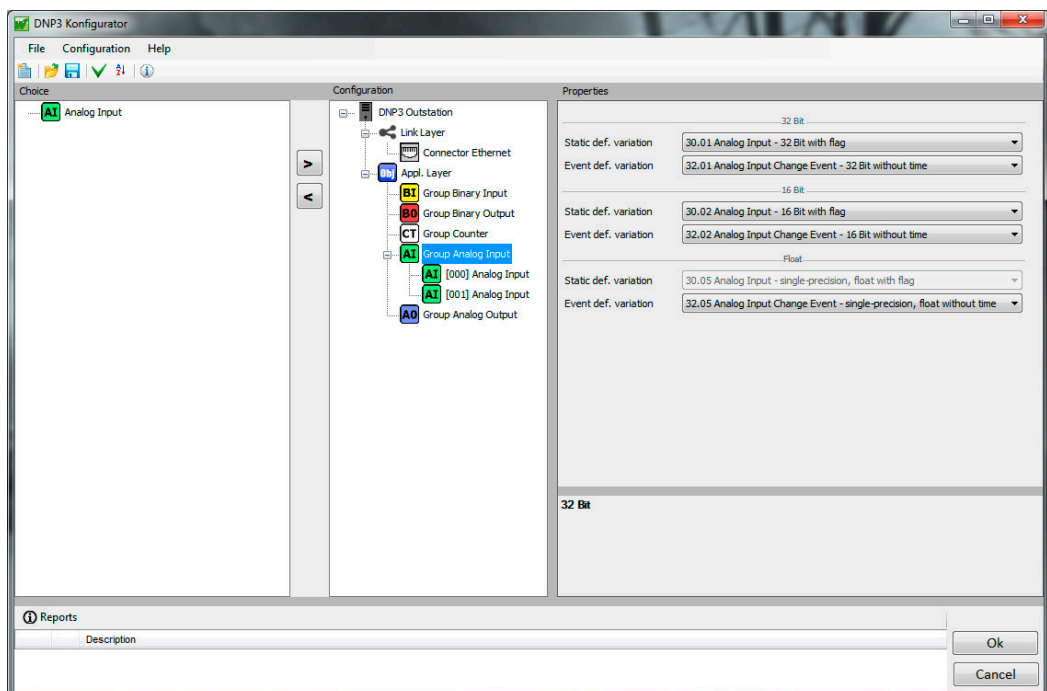


Figure 6: DNP3 Konfigurator – Add “Analog Input” Information Object

8. Click on the button [>>] to apply the information object in the “Configuration” area.

9. Click on the information object **[000] Analog Input** in the “Configuration” area.
10. Make the following settings one after the other in the “Properties” area.
 - Enter the variable name **uiRPM** in the “Variable” input field.
 - In the list next to this field, select the item **32 Bit**.

The screenshot shows the 'Properties' dialog box for an 'Analog Input' object. The 'Index' is 0. The 'Variable' field is checked and contains 'uiRPM', with a dropdown menu set to '32 Bit'. The 'Event Hysteresis' is 10. The 'Phy. Input' checkbox is unchecked, and the 'Phy. Input' field contains '%IWO'. The 'DNP3 value min.' and 'Phy. Input min.' are both 0. The 'DNP3 value max.' and 'Phy. Input Max.' are both 32767. The 'Scaling' field is 0, and the 'DNP3 value max.' field has a help icon.

Figure 7: Required Settings for the “[000] Analog Input” Information Object

11. Repeat steps 9 and 10 for the information object **[001] Analog Input**. Assign the variable name **uiVoltage**.
12. Click on **[OK]**.

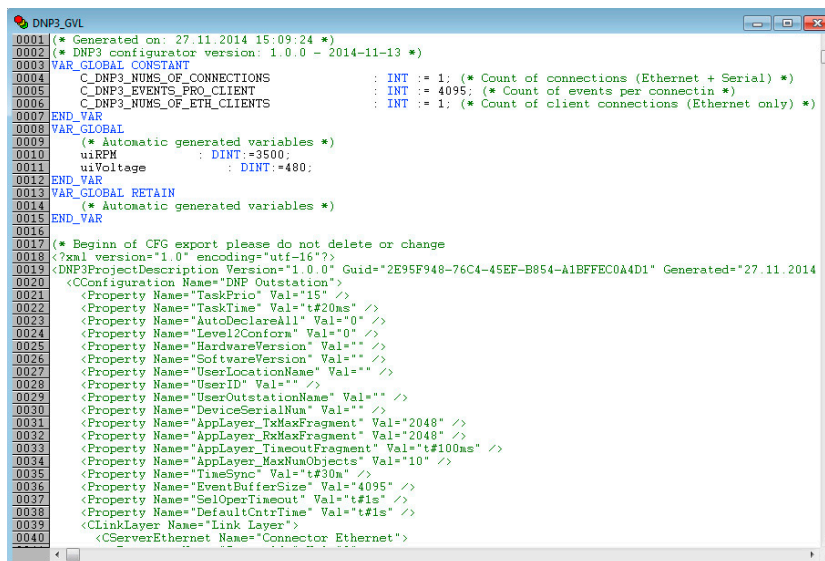
DNP3 configuration is then complete.

3.8 Open Program, Assign Values and Start Program

1. Select the **Resources** tab.
2. In the directory, select the **Global Variables** folder.
3. Select the **DNP3_GVL (CONSTANT)** entry in the tree structure.

The program editor then opens.

4. In the source text, search for the point at which the global variables **uiRPM** and **uiVoltage** are located.



```

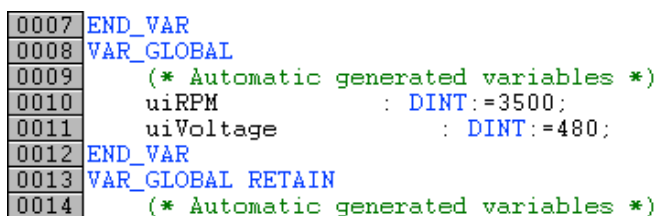
DNP3_GVL
0001 (* Generated on: 27.11.2014 15:09:24 *)
0002 (* DNP3 configurator version: 1.0.0 - 2014-11-13 *)
0003 VAR_GLOBAL CONSTANT
0004   C_DNP3_NUMS_OF_CONNECTIONS          : INT := 1; (* Count of connections (Ethernet + Serial) *)
0005   C_DNP3_EVENTS_PRO_CLIENT            : INT := 4095; (* Count of events per connectin *)
0006   C_DNP3_NUMS_OF_ETH_CLIENTS          : INT := 1; (* Count of client connections (Ethernet only) *)
0007 END_VAR
0008 VAR_GLOBAL
0009   (* Automatic generated variables *)
0010   uiRPM                               : DINT:=3500;
0011   uiVoltage                             : DINT:=480;
0012 END_VAR
0013 VAR_GLOBAL RETAIN
0014   (* Automatic generated variables *)
0015 END_VAR
0016
0017 (* Beginn of CFG export please do not delete or change
0018 <?xml version="1.0" encoding="utf-16"?>
0019 <DNP3ProjectDescription Version="1.0.0" Guid="2E95F948-76C4-45EF-B854-A1BFEC0A4D1" Generated="27.11.2014
0020 <Configuration Name="DNP Outstation">
0021   <Property Name="TaskPrio" Val="15" />
0022   <Property Name="TaskTime" Val="t#20ms" />
0023   <Property Name="AutoDeclareAll" Val="0" />
0024   <Property Name="Level2Comforn" Val="0" />
0025   <Property Name="HardwareVersion" Val="" />
0026   <Property Name="SoftwareVersion" Val="" />
0027   <Property Name="UserLocationName" Val="" />
0028   <Property Name="UserID" Val="" />
0029   <Property Name="UserOutstationName" Val="" />
0030   <Property Name="DeviceSerialNum" Val="" />
0031   <Property Name="AppLayer_TxMaxFragment" Val="2048" />
0032   <Property Name="AppLayer_RxMaxFragment" Val="2048" />
0033   <Property Name="AppLayer_TimeoutFragment" Val="t#100ms" />
0034   <Property Name="AppLayer_MaxNumObjects" Val="10" />
0035   <Property Name="TimeSync" Val="t#30m" />
0036   <Property Name="EventBufferSize" Val="4095" />
0037   <Property Name="SelOperTimeout" Val="t#1s" />
0038   <Property Name="DefaultCutTime" Val="t#1s" />
0039   <LinkLayer Name="Link Layer">
0040     <ServerEthernet Name="Connector Ethernet">

```

Figure 8: Program Editor “DNP3_GVL”

5. Assign the following values to the variables:

- **uiRPM: 3500**
- **uiVoltage: 480**



```

0007 END_VAR
0008 VAR_GLOBAL
0009   (* Automatic generated variables *)
0010   uiRPM                               : DINT:=3500;
0011   uiVoltage                             : DINT:=480;
0012 END_VAR
0013 VAR_GLOBAL RETAIN
0014   (* Automatic generated variables *)

```

Figure 9: Assigning Values in the “DNP3_GVL” Program Editor

6. Click on **[Save]**.
7. Click **Online > Login**.
8. Click **Online > Run**.

The program is then started.

3.9 Simulation of a Telecontrol Station

Note



Use the “DNP3 protocol” software as the remote station!

In this project, the “DNP3 protocol” software is used as the remote station. You can download the software free of charge from the Internet. Use the key word “DNP3 SourceForge” to search for the software.

The following steps require that you have successfully downloaded the “DNP3 protocol” software from the Internet and unpacked it.

3.10 Set the General Parameters

1. Start “DNP3 protocol”.

The “DNP Master Station” dialog appears.

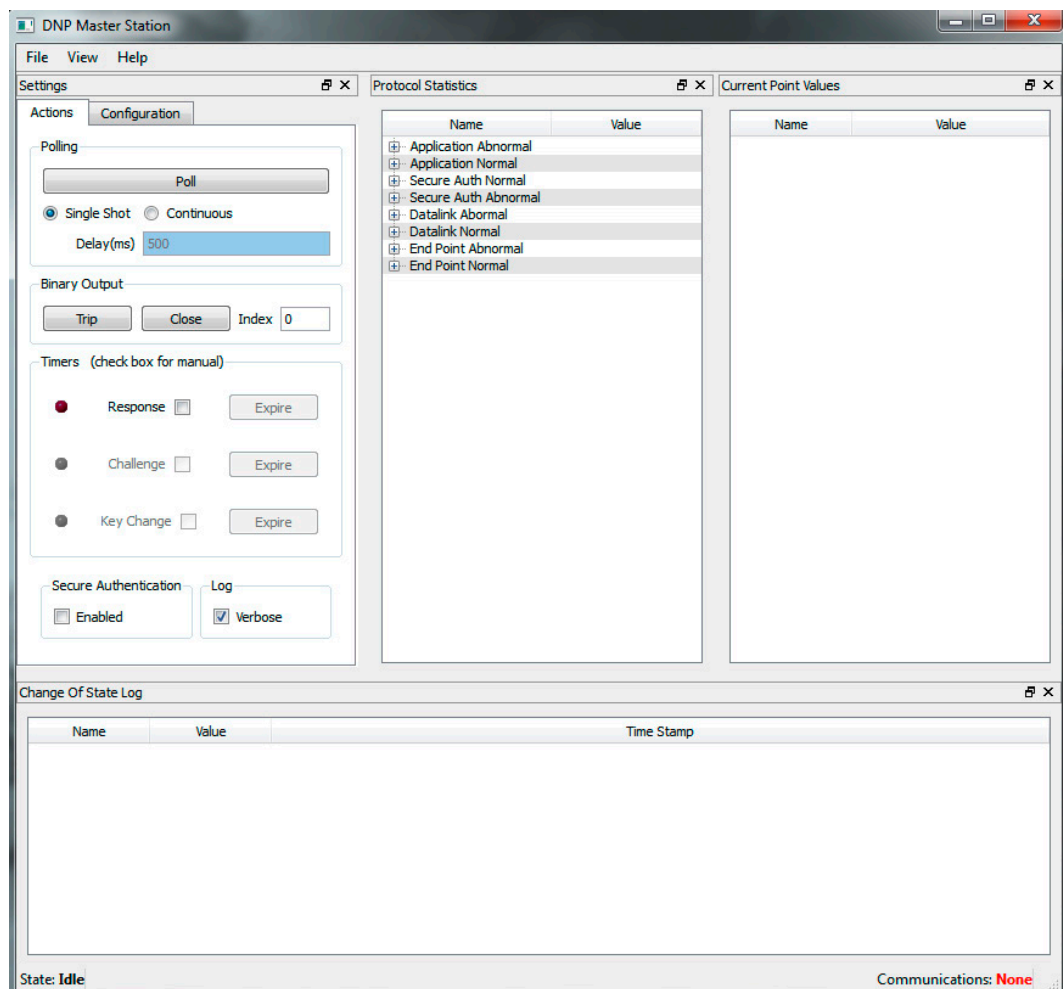
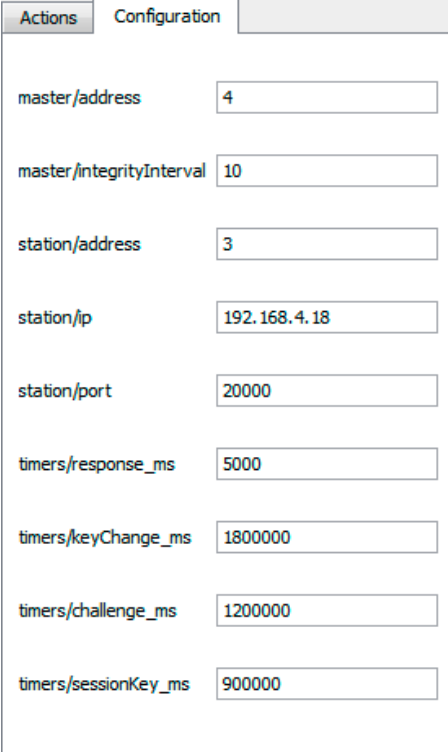


Figure 10: “DNP Master Station” Dialog

2. Click on the “Configuration” tab.

3. Make the following settings one after the other:
 - In the “master/address” input field, enter the value **4**.
 - In the “station/address” input field, enter the value **3**.
 - In the “station/ip” input field, enter the IP address **192.168.4.18**.



Field	Value
master/address	4
master/integrityInterval	10
station/address	3
station/ip	192.168.4.18
station/port	20000
timers/response_ms	5000
timers/keyChange_ms	1800000
timers/challenge_ms	1200000
timers/sessionKey_ms	900000

Figure 11: “Configuration” Tab

4. Close “DNP3 protocol”.

Any settings that have been made will be applied.

5. Start “DNP3 protocol”.
6. Click on the “Actions” tab.

Make the following settings one after the other in the “Polling” area:

- Mark the checkbox “Continuous.”
- Click the **[Poll]** button.

7. In the right tree structure click on [+] in front of the item “Analog Inputs”.

A tree structure containing two subitems then opens. These items contain the values assigned to the variables in advance:

- Subitem **STn3_Ai1**: 480
- Subitem **STn3_Ai0**: 3500

Name	Value
[-] Analog Inputs	
[-] Stn3_Ai1	480
[-] Stn3_Ai0	3500

Figure 12: Assigned Values Are Output

The telecontrol station and telecontrol substation are now communicating with one another.

4 DNP3 Configurator

Different aspects of the DNP3 Configurator are explained below. Each aspect is described in a separate subsection.

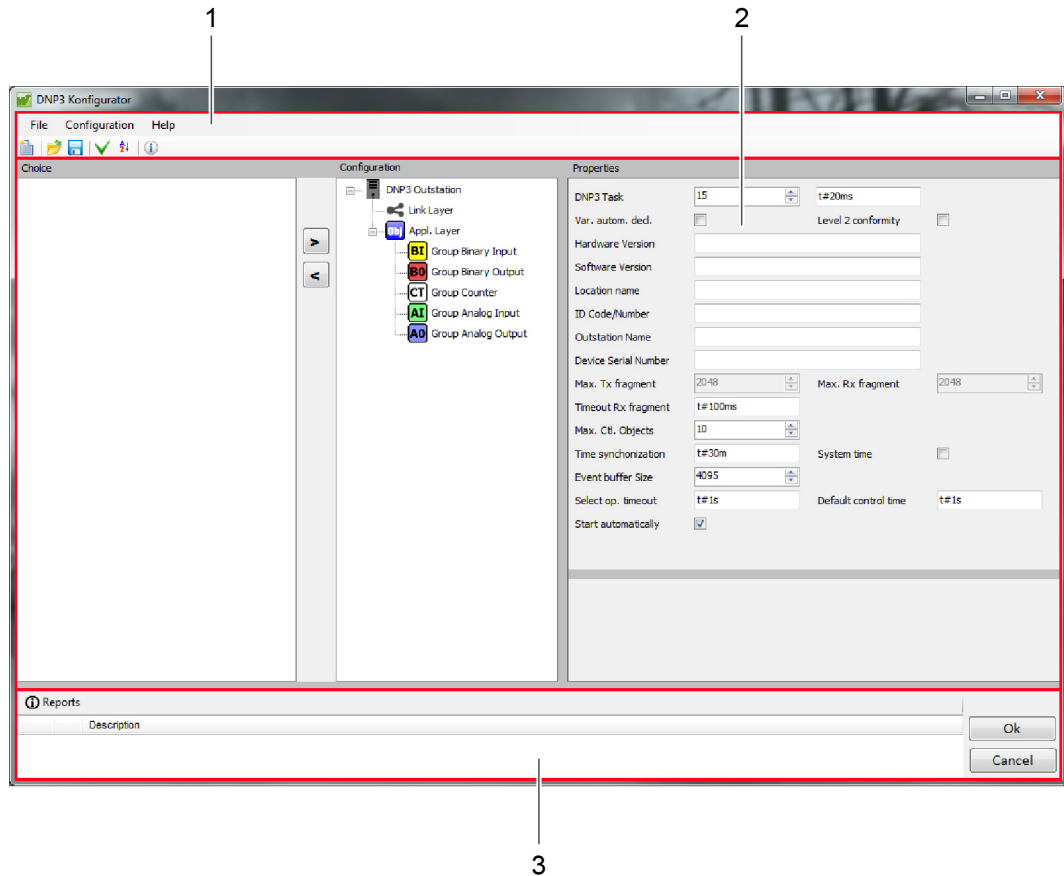


Figure 13: DNP3 Configurator Overview

Table 5: Legend for “DNP3 Configurator Overview”

Number	Area	Description
1	Menu Bar	Using the functions in the Configurator menu bar, you can export and import data. In addition, you can adjust the Configurator's display.
2	Operating range	Configuration objects and their specific information objects can be created and their parameters assigned in the workspace. The workspace is divided into the three following areas: <ul style="list-style-type: none"> • “Selection” area • “Configuration” area • “Properties” area
3	Status bar	The status bar contains buttons for closing the dialog window and status and error messages for actions that have been performed (for example <i>Import/Export file</i>).

4.1 Menu Bar

4.1.1 “File” Menu

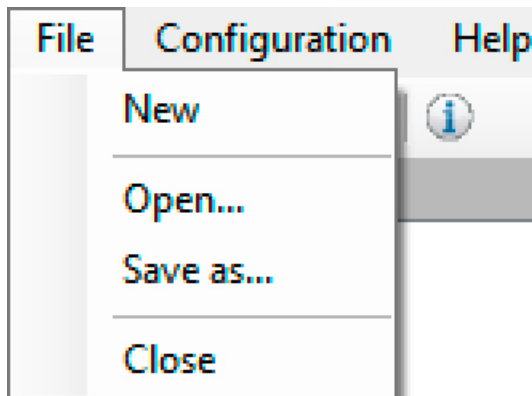


Figure 14: “File” Menu

Table 6: “File” Menu

Menu Item	Action
New	Create a new configuration.
Open...	Open an existing configuration. Three different file types are supported. The corresponding formats are described in the table “Supported Data Formats”.
Save as...	Save a configuration that has been created. Three different data types can be generated. The corresponding formats are described in the table “Supported Data Formats”.
Close	Close the configuration. If changes were made, a security question will appear asking if the affected information should be saved or discarded.

Table 7: Supported File Formats

File Format	Description
*.xml	<ul style="list-style-type: none"> • WAGO DNP3 Configurator File: XML file containing all the settings for the active configuration. • DNP3 Device Profile File: Standardized file format that enables exchange of data between different DNP3 systems.
*.csv	<ul style="list-style-type: none"> • CSV File: Data points can be exported for external processing, or externally processed data can be imported. External processing can take place in Microsoft Excel, for example. When importing data, the user can select whether the data points from the file are to replace existing data points, or whether new data points are to be added. CSV import is available solely at the group level and applies to all objects in the selected group.

4.1.2 “Configuration” Menu

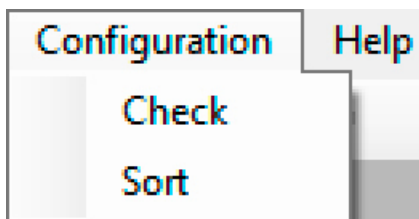


Figure 15: “Configuration” Menu

Table 8: “Configuration” Menu

Menu Item	Action
Check	The current configuration can be checked using this item. The results are displayed in the status bar at the conclusion of the check. Any errors will be highlighted.
Sort	Here, the items in the “Configuration” area can be sorted automatically according to the following criteria: <ul style="list-style-type: none"> • Groups • Variants • Information numbers

4.1.3 “Help” Menu

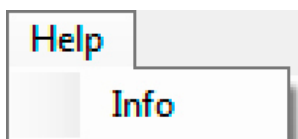


Figure 16: “Help” Menu

Table 9: “Help” Menu

Menu Item	Action
Info	The current version of the Configurator and general information about the version are displayed here.

4.2 Workspace

You can configure a substations based on DNP3 in the workspace. The workspace is divided into the three following areas:

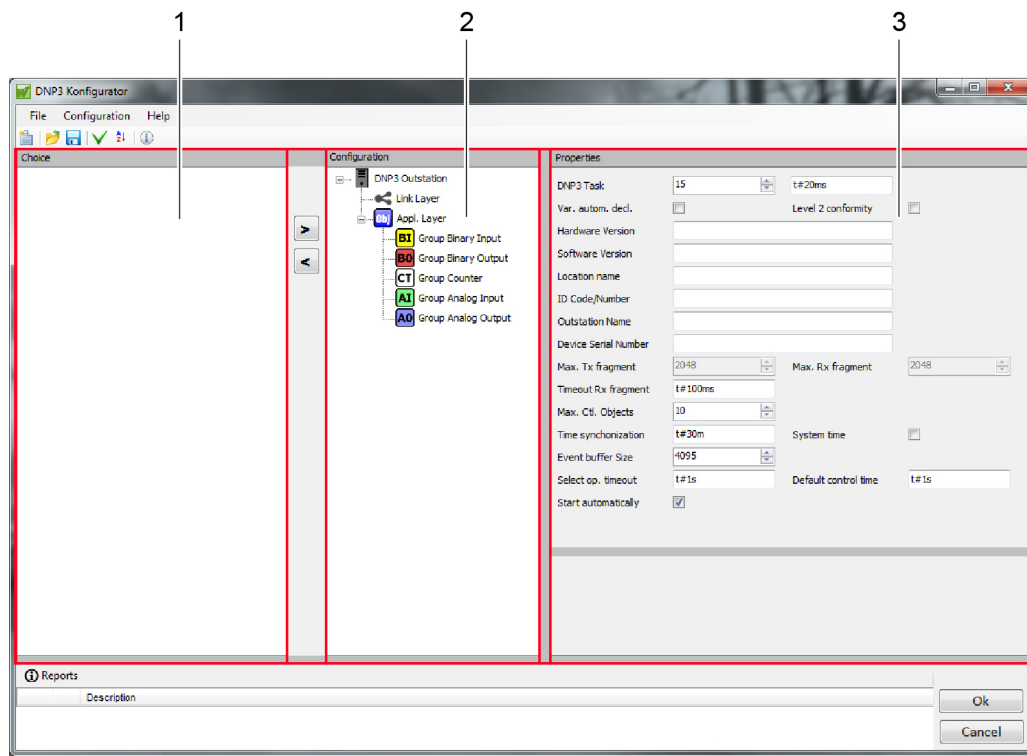


Figure 17: DNP3 Workspace Overview

Table 10: Legend for “DNP3 Workspace Overview”

Number	Area	Description
1	Choice	The information objects that can be selected for the item that was selected in the “Configuration” area are displayed in this area. Add these objects to the configuration using the [>] button.
2	Configuration	The current configuration is presented in a tree structure in this area.
3	Properties	The specific parameters for the item that was selected in the “Configuration” area can be set or changed in this area.

Various settings must be made to configure the substation for DNP3. Three different categories of parameters exist for this:

- Global parameters
- Parameters for the “Link Layer” item
- Parameters for the “Application Layer” item

More information about the individual parameter categories are given in the following sections.

Information



Specific information about a parameter!

When you move the cursor over a parameter, a message text with specific information about that parameter appears at the bottom of the dialog window!

4.2.1 Global Parameters

Global parameters can be set or changed when the item “DNP Outstation” is selected. Some of these global settings can be saved in the “DNP3 device profile file”. A text message at the bottom indicates which global parameters can be saved. Move the cursor over the parameter.

The following settings are possible:

Figure 18: Setting Options for the “DNP Outstation” Item

Table 11: Setting Options for the “DNP Outstation” Item

Setting Options	Description
DNP3 Task	The task priority and cycle time with which the generated code is processed at the target system can be specified here.
Var. autom. decl.	If this checkbox is marked, the Configurator automatically creates variables. This setting is based on the default setting for all input fields in which variables can be created.

Table 11: Setting Options for the “DNP Outstation” Item

Setting Options	Description
Level 2 conformity	If this checkbox is marked, only applications which are “DNP Level 2”-compliant can be configured. In this case, some variants cannot be used, such as “Analog Inputs” in the “FLOAT” format. Non-supported objects are not displayed in the Configurator. If the checkbox is marked for an existing configuration, the Configurator checks the objects that are used and issues a message if objects in a “REAL” format have been used. Conversion is possible when the object types have been modified in 32-bit or 16-bit objects.
Hardware Version	These parameters have no impact on the functionality of the substation. They are stored in a “DNP3 device profile file” and can be processed by other systems.
Software Version	
Location name	
ID Code/Number	
Outstation Name	
Device Serial Number	
Max. Tx fragment	This setting is permanently set to 2048 bytes/fragments and cannot be changed. This value corresponds to the maximum possible fragment size based on the standard.
Max. Rx fragment	This setting is permanently set to 2048 bytes/fragments and cannot be changed. Limiting is not required here, as shorter fragments can be easily processed.
Timeout Rx fragment	The maximum time between two link fragments accepted by the substation is entered here.
Max. Ctl. Objects	The maximum number of objects in the control direction (in one fragment) accepted by the substation is entered here.
Time synchronization	The time interval after which the substation requests clock synchronization from the master station is entered here.
System time	When this checkbox is marked, the time received from a master station is loaded to the real-time clock of the target system.
Event buffer size	The maximum number of event entries that can be stored in the substation is specified here. If events cannot be transferred to the master station and if more events occur than are specified by this parameter, any further incoming events will not be saved. Every link connection uses a separate event buffer. This option requires more memory when the number of entries is increased, or when more link connections parameters are specified.
Select op. timeout	The maximum time in which an “Operate” is accepted after a “Select” is entered here. “Select” and “Operate” are functions for writing of commands and setpoints from the master to the substation.
Confirm timeout	The substation can request an acknowledgement, for example, during event transmissions. The substation’s waiting time for the acknowledgment from the master station can be set here. If an acknowledgment is received before this time is exceeded, for example, the last transmitted events, are deleted from the event buffer. If no acknowledgment is received, the transmitted events remain in the event buffer. These events will be sent again with the next request.
Default control time	The execution time for a command can be specified here. This time is only used when an explicit execution time has been defined for a command from the master station.
Start automatically	The DNP3 application is started automatically when this checkbox is marked. The DNP3 application will not start automatically if this checkbox is not marked. The application can only be started manually using the variable <code>gxDNP3_Enable</code> . This provides the application with a non-specified time to read in configuration from files for example. Internal initialization does not begin until the variable <code>gxDNP3_Enable</code> is set to TRUE .

4.2.2 “Link Layer” Parameters

Specific parameters can be set or changed for every object included under the "Link Layer" item. Using these specific parameters, properties can be defined for setting up a connection between the substation and the master station.

4.2.2.1 “Connector Ethernet” Object

These specific parameters are used to set up an ETHERNET connection between the substation and the master station. Up to four connections can be configured, depending on the target system.

The screenshot shows the 'Properties' dialog for a 'Connector Ethernet' object. The fields are as follows:

- Slave Addr.: 1
- Network Type: TcpListening / Udp
- Network Port TCP: 20000
- Network Port UDP: 20000
- Own IP: 192.168.10.10
- Client connections: 1 Client
- TCP Keep Alive [s]: 10
- IP Filter:
- Address validate:
- Master Addr.: 2
- Permitted IP 1: 192.168.2.1
- Permitted IP 2: 0.0.0.0
- Permitted IP 3: 0.0.0.0
- Master Addr.: 3
- Permitted IP 1: 192.168.2.2
- Permitted IP 2: 0.0.0.0
- Permitted IP 3: 0.0.0.0
- Master Addr.: 4
- Permitted IP 1: 192.168.2.3
- Permitted IP 2: 0.0.0.0
- Permitted IP 3: 0.0.0.0
- Master Addr.: 5
- Permitted IP 1: 192.168.2.4
- Permitted IP 2: 0.0.0.0
- Permitted IP 3: 0.0.0.0

Figure 19: Setting Options for a “Connector Ethernet” Object

Table 12: Setting Options for a “Connector Ethernet” Object

Setting Options	Description
Slave Addr.	The address for the link layer (substation) is specified here. This address is used in the link layer telegram to coordinate the source and the target.
Network Type	The “TCPListening/UDP” setting is specified by the master station and cannot be changed.
Netzwerk Port TCP	The port that is to be used for the connection is specified here. The standard port for DNP3 is 20000.
TCP Keep Alive [s]	DNP3 uses the “Keep Alive” mechanism to monitor the status of a connection. Here, you can define the time after which a keep-alive telegram is expected and sent by the substation.
Netzwerk Port UDP	The port that is to be used for the connection is specified here. The standard port for DNP3 is 20000.

Table 12: Setting Options for a “Connector Ethernet” Object

Setting Options	Description
Own IP	The device's own IP address does not have any effect on the functionality of the substation. It is stored in a “DNP3 device profile file” and can be processed by other systems.
Client connections	The number of client connections can be specified here.
IP Filter	If this checkbox is marked, the IP address which are accepted when setting up a connection can be restricted.
Address validate	If this checkbox is marked, the master address of the incoming telegram is evaluated. If the master address and the link layer (see input field “Master Addr.”) are identical, the incoming telegram is processed. If this checkbox is not marked, the master address of the incoming telegram will be ignored. However, the telegram is evaluated.
Master Addr.	The address for the link layer (master station) is specified here. This address is used in the link layer telegram to coordinate the source and the target.
Permitted IP1 ... Permitted IP3	IP addresses which are accepted when setting up a connection can be specified here.

4.2.2.2 “Connector Serial” Object

These specific parameters are used to set up a serial connection between the substation and the master station. Up to four connections can be configured, depending on the target system. Every connection must also have its own physical interface.

The screenshot shows a 'Properties' window with the following settings:

- Slave Addr.: 1
- Master Addr.: 2
- Interface: COM 0
- Baudrate: 9600
- Transmission type: RS 232

Figure 20: Setting Options for a “Connector Serial” Object

Table 13: Setting Options for a “Connector Serial” Object

Setting Options	Description
Slave Addr.	The address for the link layer (substation) is specified here. This address is used in the link layer telegram to coordinate the source and the target.
Master Addr.	The address for the link layer (master station) is specified here. This address is used in the link layer telegram to coordinate the source and the target.
Interface	Here, you can select the interface that is to be used in line with the given target system and the fieldbus nodes. Allocation of the individual interfaces to the given target system is shown in the appendix.
Baudrate	The baud rate for the 750-652 “Serial interface” I/O module can be selected here. The selected baud rate should not exceed 19200 baud for this.
Transmission type	This setting is permanently set to the RS-232 transmission type and cannot be changed. Note: Transfer type RS-485 cannot be selected. Collision detection, which is not available in the target system, must be provided on the bus to ensure and maintain communication.

4.2.3 “Application Layer” Parameters

The group objects included under the “Application Layer” item form a structure containing the corresponding information objects. These information objects can be added once or several times to the individual group objects.

Specific parameters can be set or changed for every group object and for each corresponding information object.

Information



Table containing information objects and variations!

A table containing all of the information objects and variations supported for DNP3 is given in the appendix!

Note



Indices (object addresses) may be shifted during sorting!

All of the information objects are sorted before the configuration is compiled or before the menu item **Configuration > Sort** is selected. Indices can be shifted during sorting, as the data types are also taken into account for sorting!

4.2.3.1 “Group Binary Input” Group Object

Properties	
Static def. variation	01.02 Binary Input - With flag
Event def. variation	02.02 Binary Input Event - With absolute time

Figure 21: Setting Options for the “Group Binary Input” Group Object

Table 14: Setting Options for the “Group Binary Input” Group Object

Setting Options	Description
Static def. variation	<p>Here, you can select the format for the response to a data request if no defined telegram format has been defined when a request from the master station is present.</p> <p>The following can be selected for this:</p> <p>01.01 Binary Input – Packed format <i>(1 bit per input)</i> 01.02 Binary Input – With flag <i>(1 byte per input)</i></p>
Event def. variation	<p>Here, you can select whether an event message is to be transmitted and the time stamp to be used if no setting has been defined for event messages for a request from the master station.</p> <p>The following can be selected for this:</p> <p>02.01 Binary Input Event – Without time 02.02 Binary Input Event – With absolute time 02.03 Binary Input Event – With relative time</p>

Information Object "Binary Input"

As a rule, 8 inputs are commonly parameterized. These inputs are always assigned to the same event category.

The screenshot shows the 'Properties' dialog for a 'Binary Input' object. At the top, the title is 'Binary Input'. Below it, the 'Index' is set to 0. Under 'Binary Input', there are two options: '[1] Static' which is checked with a checkbox, and '[2] Event' which is set to 'No Event' in a dropdown menu. Below this is a section titled 'Binary input block' containing a table with three columns: 'Autom. declare', 'Name', and 'Description'. The table has eight rows, one for each bit from 'Bit 0:' to 'Bit 7:'. Each row has a checkbox in the 'Autom. declare' column, and empty text boxes for 'Name' and 'Description'.

Figure 22: Setting Options for a “Binary Input” Information Object

Table 15: Setting Options for a “Binary Input” Information Object

Setting Options	Description
Index	The index corresponds to the object address. This is a permanent setting and is counted in ascending order for every item. The Configurator assigns the next available index automatically.
Binary Input (1) Static	This checkbox is marked permanently and indicates that the master station can query the current status at all times.
Binary Input (2) Event	The event category is selected according to DNP3. An event is generated as soon as a different signal is present at the inputs and this event saved to the event buffer with its event category. Events are always transferred and saved with a time flag.
Input fields for Bit 0 ... Bit 7:	
Autom. declare	If the checkbox is marked, the specified variable will be created automatically in a global list. Exception: All addresses based on IEC 61131 which begin with the character “%” will not be created.
Variable	A variable name based on IEC 61131 must be entered here. An I/O address, which is used as a signal source can also be entered here as an option.
Name	These parameters have no impact on the functionality of the substation. They are stored in a “DNP3 device profile file” and can be processed by other systems.
Description	

4.2.3.2 “Group Binary Output” Group Object

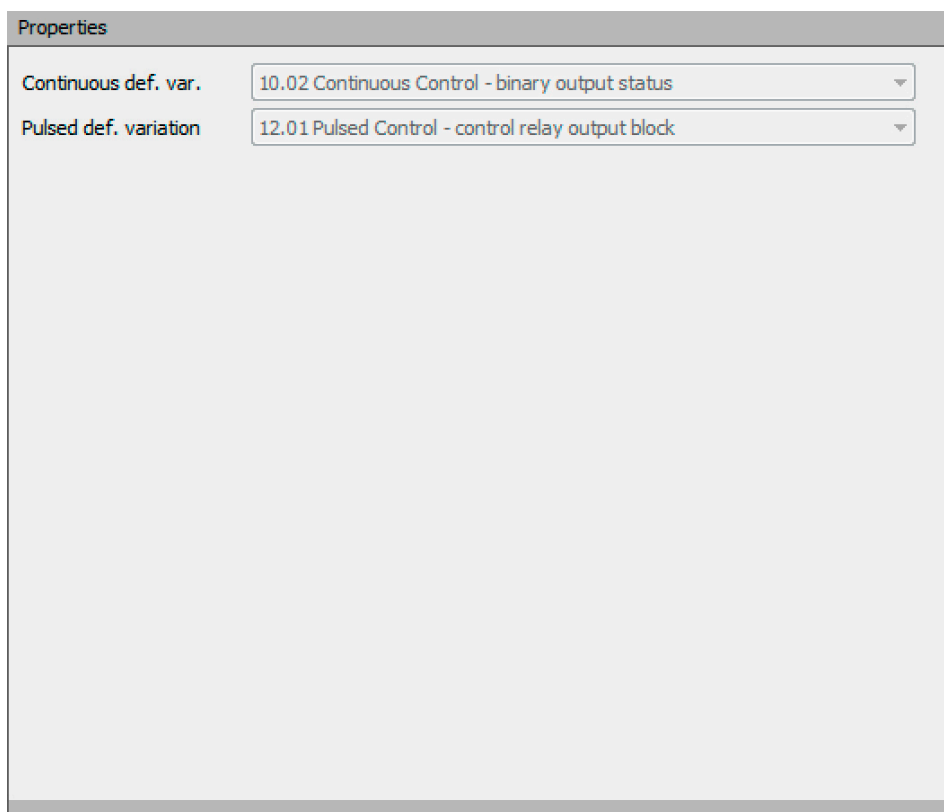


Figure 23: Setting Options for the “Group Binary Output” Group Object

Table 16: Setting Options for the “Group Binary Output” Group Object

Setting Options	Description
Continuous def. var.	<p>Here, you can select the format for the response to a data request if no defined telegram format has been defined when a request from the master station is present.</p> <p>The following fixed setting is given for this:</p> <p>10.02 Continuous Control – binary output status</p>
Pulsed def. variation	<p>Here, you can select the data format in which an event message is to be transmitted if no setting has been defined for event messages for a request from the master station.</p> <p>The following fixed setting is given for this:</p> <p>12.01 Pulsed Control – control relay output block</p>

Information Object “Binary Output”

Figure 24: Setting Options for a “Binary Output” Information Object

Table 17: Setting Options for a “Binary Output” Information Object

Setting Options	Description
Index	The index corresponds to the object address. This is a permanent setting and is counted in ascending order for every item. The Configurator assigns the next available index automatically.
Name	These parameters have no impact on the functionality of the substation. They are stored in a “DNP3 device profile file” and can be processed by other systems.
Description	
Binary Output (10) Static	This checkbox is marked permanently and indicates that the master station can query the current status at all times.
Binary Output (12) Command	This checkbox is marked permanently and indicates that commands which control the outputs are supported.
Control model	<p>Here, you can select two different control models according to DNP3 which regulate the response of the digital outputs when they are triggered by the master station.</p> <p>The following can be selected for this:</p> <p>“Activation model”: The digital outputs transmit pulse sequences or pulses with definable On/Off times or repetition.</p> <p>“Complementary latch model”: The master station switches the digital output on or off in a defined manner.</p> <p>“Retain” checkbox: This checkbox is shown when “Complementary latch model” has been selected. If the checkbox is marked, the status of the digital output remains stored after a restart.</p>

Table 17: Setting Options for a “Binary Output” Information Object

Setting Options	Description
Variable	<p>The following settings are possible:</p> <p>Checkbox: If the checkbox is marked, the specified variable will be created automatically in a global list. Exception: All addresses based on IEC 61131 which begin with the character “%” will not be created.</p> <p>Input Field: A variable name based on IEC 61131 must be entered here. An I/O address, which is used as a signal source can also be entered here as an option.</p>

4.2.3.3 “Group Counter” Group Object

There are two different types of counters for DNP3:

- 32-bit counters
- 16-bit counters

Specific setting options for the default variations exist for each type of counter in the “Properties” area.

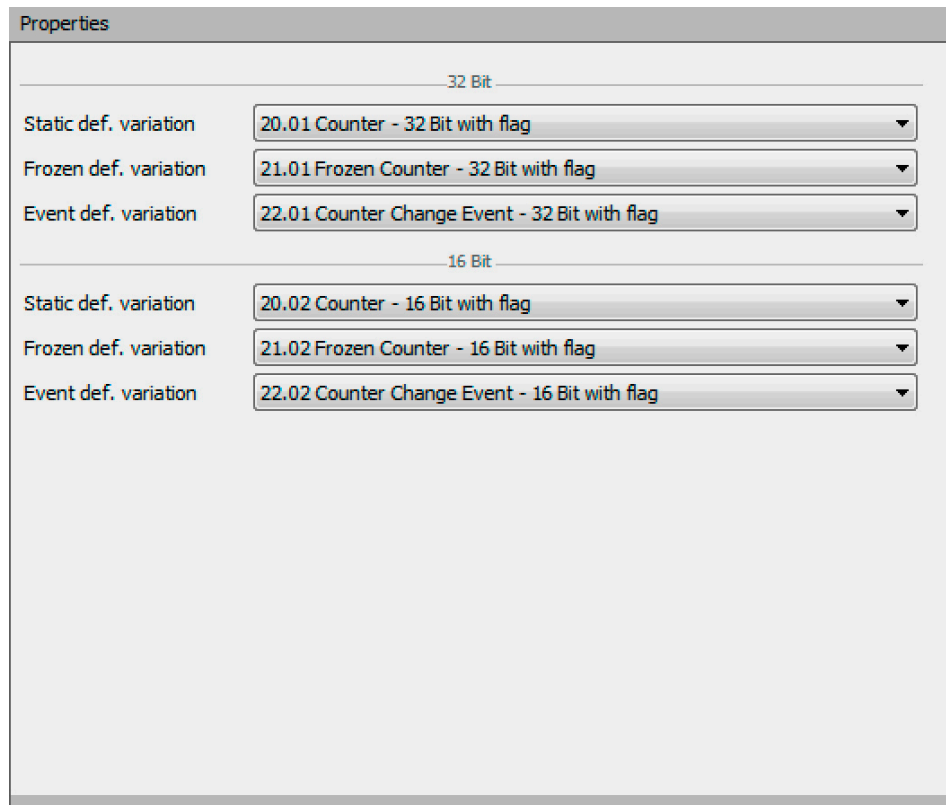


Figure 25: Setting Options for the “Group Counter” Group Object

Table 18: Setting Options for the “Group Counter” Group Object

Setting Options	Description
Static def. variation	<p>Here, you can select the volume of data to be transferred if no defined telegram format is provided for a request from the master station.</p> <p>The following options are available for a 32-bit counter:</p> <p>20.01 Counter – 32 Bit with flag 20.05 Counter – 32 Bit without flag</p> <p>The following options are available for a 16-bit counter:</p> <p>20.02 Counter – 16 Bit with flag 20.06 Counter – 16 Bit without flag</p>

Table 18: Setting Options for the “Group Counter” Group Object

Setting Options	Description
Frozen def. variation	<p>Here, you can select the volume of frozen values to be transferred if no setting has been defined for frozen values for a request from the master station.</p> <p>The following options are available for a 32-bit counter: 21.01 Frozen Counter – 32 Bit with flag 21.05 Frozen Counter – 32 Bit with flag and time 21.09 Frozen Counter – 32 Bit without flag</p> <p>The following options are available for a 16-bit counter: 21.02 Frozen Counter – 16 Bit with flag 21.06 Frozen Counter – 16 Bit with flag and time 21.10 Frozen Counter – 16 Bit without flag</p>
Event def. variation	<p>Here, you can select the format in which an event message is to be transmitted if no setting has been defined for event messages for a request from the master station.</p> <p>The following fixed setting is given for the 32-bit counter: 22.01 Counter Change Event – 32 Bit with flag</p> <p>The following fixed setting is given for the 16-bit counter: 22.02 Counter Change Event – 16 Bit with flag</p>

Information Object “Counter”

The screenshot shows the configuration interface for a Counter object. The fields are as follows:

- Index:** A text input field containing the value '0'.
- Name:** An empty text input field.
- Description:** An empty text input field.
- Counter:** A row with two settings: '[20] Static' with a checked checkbox, and '[22] Event' with a dropdown menu currently set to 'No Event'.
- Frozen Counter:** A row with '[21] Static' and a checked checkbox.
- Variable:** A row with an unchecked checkbox, an empty text input field, a dropdown menu set to 'Edge (32 Bit Count)', and an unchecked checkbox labeled 'Retain'.
- Event Hysteresis:** A text input field containing the value '10'.

Figure 26: Setting Options for a “Counter” Information Object

Table 19: Setting Options for a “Counter” Information Object

Setting Options	Description
Index	The index corresponds to the object address. This is a permanent setting and is counted in ascending order for every item. The Configurator assigns the next available index automatically.
Name	These parameters have no impact on the functionality of the substation. They are stored in a “DNP3 device profile file” and can be processed by other systems.
Description	
Counter (20) Static	This checkbox is marked permanently and indicates that the master station can query the current status at all times.
Counter (22) Event	The event category is selected according to DNP3. As soon as the counter value exceeds the set event hysteresis (see “Event Hysteresis” setting option), the counter value is saved with the event category in the event buffer and can then be queried asynchronously.
Frozen Counter (21) Static	This checkbox is marked permanently and indicates that the master station can query the frozen status at all times.
Variable	<p>The following settings are possible:</p> <p>Checkbox: If the checkbox is marked, the specified variable will be created automatically in a global list. Exception: All addresses based on IEC 61131 which begin with the character “%” will not be created.</p> <p>Input Field: A variable name based on IEC 61131 must be entered here. An I/O address, which is used as a signal source can also be entered here as an option.</p> <p>Selection Box: A total of three different variants is supported for this counter:</p> <p>Edge (32 Bit Counter): Variable: BOOL The positive edges are added internally for the counter value.</p> <p>32 Bit: Variable: DINT The positive edges are added externally (for example in a CODESYS application) for the counter value.</p> <p>16 Bit: Variable: INT The positive edges are added externally (for example in a CODESYS application) for the counter value.</p> <p>Checkbox “Retain”: If this checkbox is marked, “retain” variables will be created for the counter to ensure that the counter values are saved on a restart. If this checkbox is not marked, the counter values are initially set at "0" when the device is switched on.</p>
Event Hysteresis	The hysteresis can be set here. This selection field is only activated when an event category has been assigned to the counter. The counter value, along with the event category, is stored in the event buffer as soon as the set value is exceeded.

4.2.3.4 “Group Analog Input” Group Object

In DNP3 there are two different types of whole number analog values and two different types of floating point values:

- 32-bit analog values
- 16-bit analog values
- 32-bit floating point values
- 64-bit* floating point values

Specific setting options for the default variations exist for the first three types cited above in the “Properties” area.

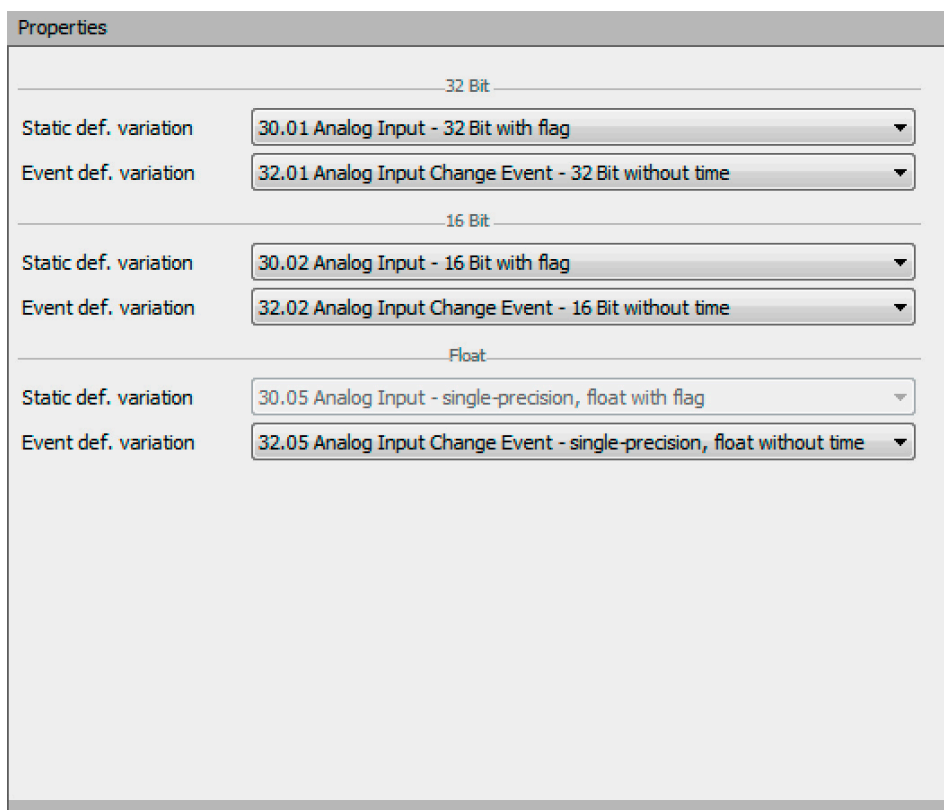


Figure 27: Setting Options for the “Group Analog Input” Group Object

Table 20: Setting Options for the “Group Analog Input” Group Object

Setting Options	Description
Static def. variation	<p>Here, you can select the format for the response to a data request if no defined telegram format has been defined when a request from the master station is present.</p> <p>The following can be selected for this (32-bit):</p> <p>30.01 Analog Input – 32 Bit with flag 30.03 Analog Input – 32 Bit without flag</p>

Table 20: Setting Options for the “Group Analog Input” Group Object

Setting Options	Description
	<p>The following can be selected for this (16-bit):</p> <p>30.02 Analog Input – 16 Bit with flag 30.04 Analog Input – 16 Bit without flag</p> <p>The following fixed setting is given for this (FLOAT):</p> <p>30.05 Analog Input – single-precision, floating-point with flag</p>
Event def. variation	<p>Here, you can select whether a reply is to be made with or without a time flag to an event request if no setting has been defined for event messages on a request from the master station.</p> <p>The following can be selected for this (32-bit):</p> <p>32.01 Analog Input Change Event – 32 Bit without time 32.03 Analog Input Change Event – 32 Bit with time</p> <p>The following can be selected for this (16-bit):</p> <p>32.02 Analog Input Change Event – 16 Bit without time 32.04 Analog Input Change Event – 16 Bit with time</p> <p>The following can be selected for this (FLOAT):</p> <p>32.05 Analog Input Change Event – single-precision, floating-point without time 32.07 Analog Input Change Event – single-precision, floating-point with time</p>

* This floating point value is currently not supported, as CODESYS 2.3 does not support any 64-bit values in the “REAL” format.

Information Object “Analog Input”

Figure 28: Setting Options for an “Analog Input” Information Object

Table 21: Setting Options for an “Analog Input” Information Object

Setting Options	Description
Index	The index corresponds to the object address. This is a permanent setting and is counted in ascending order for every item. The Configurator assigns the next available index automatically.
Name	These parameters have no impact on the functionality of the substation. They are stored in a “DNP3 device profile file” and can be processed by other systems.
Description	
Analog Input (30) Static	This checkbox is marked permanently and indicates that the master station can query the current status at all times.
Analog Input (32) Event	The event category is selected according to DNP3. As soon as the value exceeds the set event hysteresis (see “Event Hysteresis” setting option), the value is saved with the event category in the event buffer and can then be queried asynchronously.
Variable	<p>The following settings are possible:</p> <p>Checkbox: If the checkbox is marked, the specified variable will be created automatically in a global list. Exception: All addresses based on IEC 61131 which begin with the character “%” will not be created.</p> <p>Input Field: A variable name based on IEC 61131 must be entered here. An I/O address, which is used as a signal source can also be entered here as an option.</p>

Table 21: Setting Options for an “Analog Input” Information Object

Setting Options	Description
	<p>Selection Box: A total of three different variants is supported for this counter:</p> <p>32 Bit: Variable: DINT The positive edges are added externally (for example in a CODESYS application) for the counter value.</p> <p>16 Bit: Variable: INT The positive edges are added externally (for example in a CODESYS application) for the counter value.</p> <p>Float: Variable: BOOL The positive edges are added internally for the counter value.</p>
Event Hysteresis	The hysteresis can be set here. This selection field is only activated when an event category has been assigned to the analog output. The value, along with the event category, is stored in the event buffer as soon as the set value is exceeded.
Phy. Input	<p>The following settings are possible:</p> <p>Checkbox: If this checkbox is marked, the entire lower input range is enabled. If the checkbox is not marked, the variable declared in the “Variable” input field is applied.</p> <p>Input Field: A variable name based on IEC 61131 must be entered here. An I/O address, which is used as a signal source can also be entered here as an option.</p>
DNP3 value min.	<p>The value that corresponds to the physical minimum value is specified here.</p> <p>Example: 4096 (4 mA) corresponds to -20 °C. Value to enter: -20.0</p>
DNP3 value max.	<p>The value that corresponds to the physical maximum value is specified here.</p> <p>Example: 32767 (20 mA) corresponds to 60 °C. Value to enter: 60.0</p>
Phy. Input min.	<p>The input value that corresponds to the minimum process value is specified here.</p> <p>Example: 4096 corresponds to 4 mA. Value to enter: 4096</p>
Phy. Input max.	<p>The input value that corresponds to the maximum process value is specified here.</p> <p>Example: 32767 corresponds to 20 mA. Value to enter: 32767</p>
Scaling	The offset and conversion factor which the Configurator has calculated from the values input for “DNP Value Min./Max.” and “Phy.Input Min./Max.” are displayed here.

Information



Example of scaling using the icon [Help]

If the checkbox for the physical input is marked, a scaling example can be called up using the [Help] icon.

Schematic signal flow diagram for the “Analog Input” information object

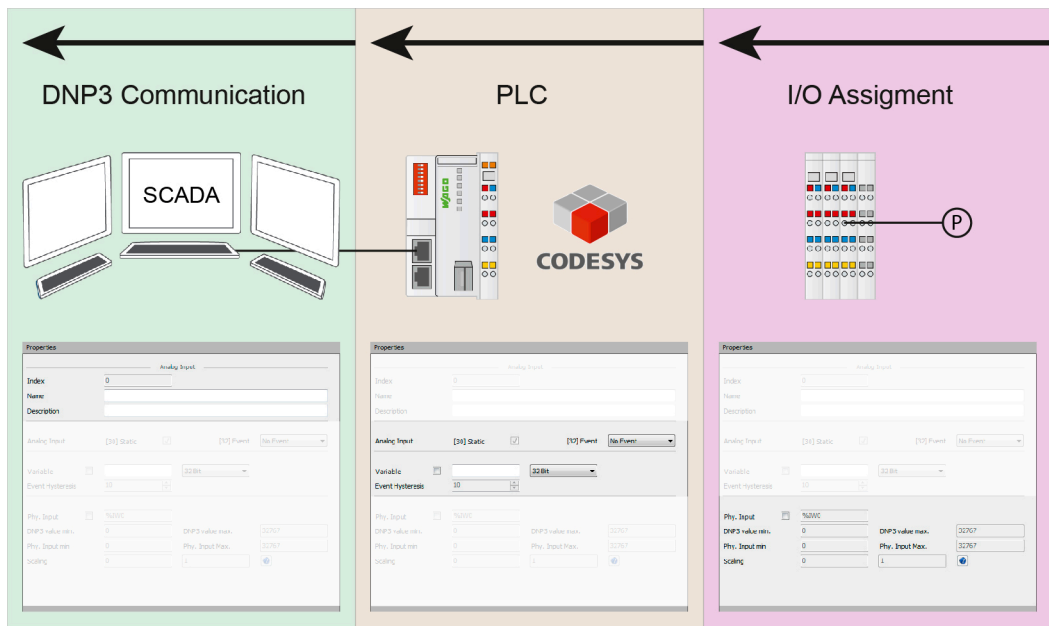


Figure 29: Schematic Signal Flow Diagram for “Analog Input” Information Object

4.2.3.5 “Group Analog Output” Group Object

In DNP3 there are two different types of whole number analog values and two different types of floating point values:

- 32-bit analog values
- 16-bit analog values
- 32-bit floating point values
- 64-bit* floating point values

Specific setting options for the default variations exist for the first three types cited above in the "Properties" area.

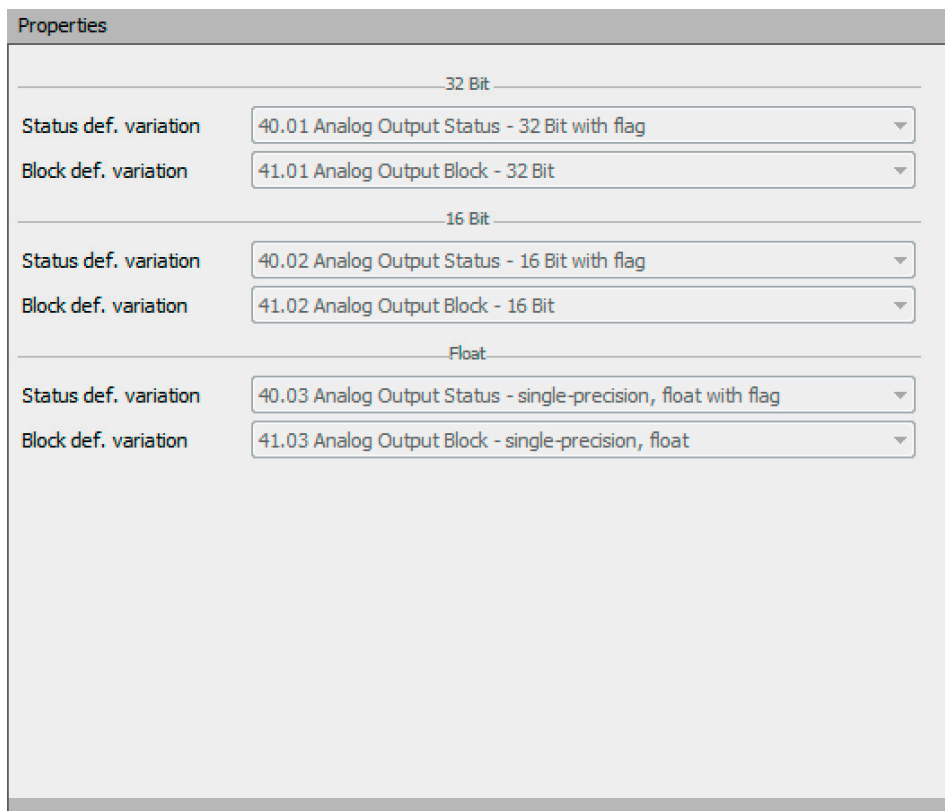


Figure 30: Setting Options for the “Group Analog Output” Group Object

Table 22: Setting Options for the “Group Analog Output” Group Object

Setting Options	Description
Status def. variation	<p>Here, you can select the format for the response to a data request if no defined telegram format has been defined when a request from the master station is present.</p> <p>The following fixed setting is given for this (32 bit):</p> <p>40.01 Analog Output Status – 32 Bit with flag</p> <p>The following fixed setting is given for this (16 bit):</p> <p>40.02 Analog Output Status – 16 Bit with flag</p>

Table 22: Setting Options for the “Group Analog Output” Group Object

Setting Options	Description
	<p>The following fixed setting is given for this (FLOAT):</p> <p>40.03 Analog Output Status – single-precision, floating-point with flag</p>
Block def. variation	<p>Here, you can select the format for the response to the data request if no setting has been defined for event messages for a request from the master station.</p> <p>The following fixed setting is given for this (32 bit):</p> <p>41.01 Analog Output Block – 32 Bit</p> <p>The following fixed setting is given for this (16 bit):</p> <p>41.02 Analog Output Block – 16 Bit</p> <p>The following fixed setting is given for this (FLOAT):</p> <p>41.03 Analog Output Block – single-precision, floating-point</p>

* This floating point value is currently not supported, as CODESYS 2.3 does not support any 64-bit values in the “REAL” format.

Information Object “Analog Output”

Figure 31: Setting Options for an “Analog Output” Information Object

Table 23: Setting Options for an “Analog Output” Information Object

Setting Options	Description
Index	The index corresponds to the object address. This is a permanent setting and is counted in ascending order for every item. The Configurator assigns the next available index automatically.
Name	These parameters have no impact on the functionality of the substation. They are stored in a “DNP3 device profile file” and can be processed by other systems.
Description	
Analog Output (40) Static	This checkbox is marked permanently and indicates that the master station can query the current status at all times.
Analog Output (41) Command	This checkbox is marked permanently and indicates that the command direction is supported at all times for analog outputs.
Variable	<p>The following settings are possible:</p> <p>Checkbox: If this checkbox is marked, the variable that has been entered will be created automatically in a global list. Exception: All addresses based on IEC 61131 which begin with the character “%” will not be created.</p> <p>Input Field: A variable name based on IEC 61131 must be entered here. An I/O address, which is used as a signal source can also be entered here as an option.</p>

Table 23: Setting Options for an “Analog Output” Information Object

Setting Options	Description
	<p>Selection Box: A total of three different variants is supported for this counter:</p> <p>16 Bit: Variable: INT The positive edges are added externally (for example in a CODESYS application) for the counter value.</p> <p>32 Bit: Variable: DINT The positive edges are added externally (for example in a CODESYS application) for the counter value.</p> <p>Float: Variable: BOOL The positive edges are added internally for the counter value.</p> <p>Checkbox “Retain”: If the checkbox is marked, the status of the analog output remains stored after a restart.</p>
Phy. Output	<p>The following settings are possible:</p> <p>Checkbox: If this checkbox is marked, the entire lower input range is enabled. If the checkbox is not marked, the variable declared in the “Variable” input field is applied.</p> <p>Input Field: A variable name based on IEC 61131 must be entered here. An I/O address, which is used as a signal source can also be entered here as an option.</p>
DNP3 value min.	<p>The value that corresponds to the physical minimum value is specified here. Example: 4096 (4 mA) corresponds to 0 min⁻¹. Value to enter: -0.0</p>
DNP3 value max.	<p>The value that corresponds to the physical maximum value is specified here. Example: 32767 (20 mA) corresponds to 3000 min⁻¹. Value to enter: 3000.0</p>
Phy. Output min.	<p>The output value that corresponds to the minimum process value is specified here. Example: 4096 corresponds to 4 mA. Value to enter: 4096</p>
Phy. Output max.	<p>The output value that corresponds to the maximum process value is specified here. Example: 32767 corresponds to 20 mA. Value to enter: 32767</p>
Scaling	<p>The offset and conversion factor which the Configurator has calculated from the values input for “DNP Value Min./Max.” and “Phy.Output Min./Max.” are displayed here.</p>

Information



Example of scaling using the icon [Help]

If the checkbox for the physical output is marked, a scaling example can be called up using the [Help] icon.

Schematic signal flow diagram for the “Analog Output” information object

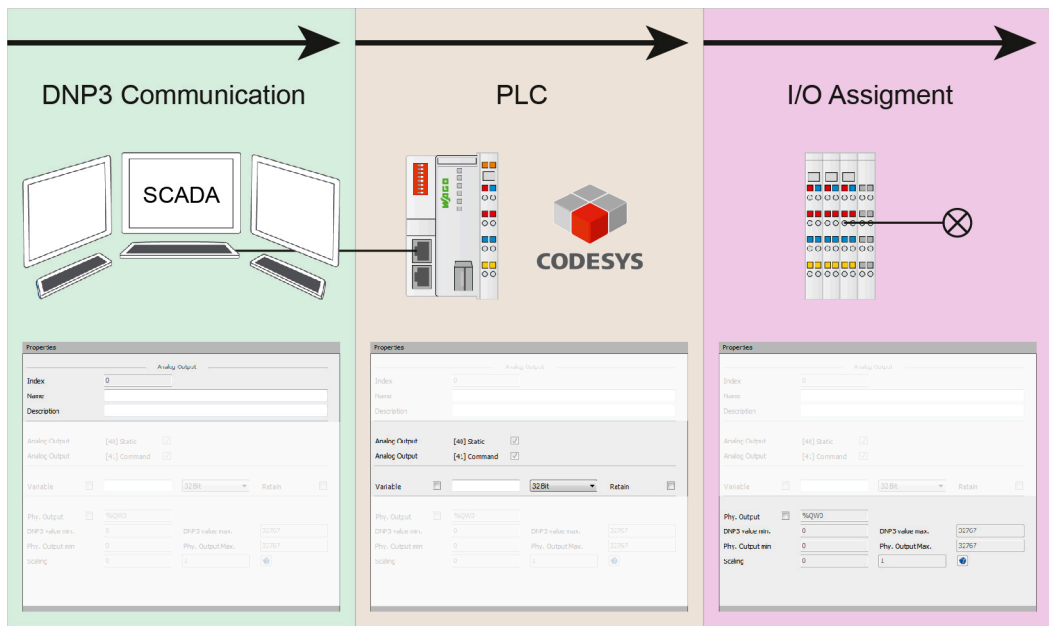


Figure 32: Schematic Signal Flow Diagram for “Analog Output” Information Object

4.3 Status Bar

Error messages are displayed in the status bar. The various error messages are listed below:

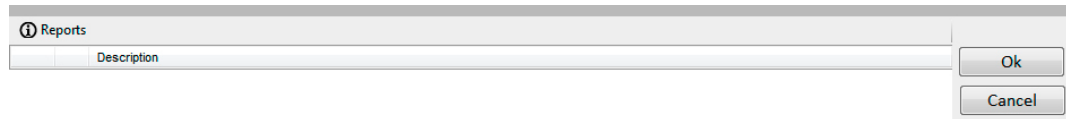


Figure 33: Status bar



Information

Navigate to the cause of the error!

When you double-click on an error message the Configurator moves automatically to the information object containing the error. This greatly facilitates troubleshooting!

Table 24: Error Messages in the Status Bar

Error message	Description
Error during import. Incorrect file format.	An attempt was made to import an XML file which did not have the correct format.
Error: At least one communication channel must be available. Add an ETHERNET or a serial connector.	No communication channel available. A connection cannot be set up, meaning that a valid configuration cannot be created. Add at least one interface and configure it as required for your connection.
Error: At least one object must be available. Add an object to the configuration.	No information object available. No valid data model can be generated. Therefore, add at least one information object to the configuration and set its parameters according to your data point list.
Error: When the physical input/output option is used, it is not possible to define a process variable as a "Variable".	Remove the variable entry and ensure that a valid address is given for the process image in the "Phy.Input/Phy. Output" input field.
Error: Minimum value of the DNP3 process value is \geq Maximum value of the DNP3 process value.	The minimum and maximum value for scaling is checked for the analog values. The minimum value of the DNP3 process value may not be greater than or equal to the maximum DNP3 process value.
Error: The physical input '###1' cannot be used for an "Analog Output".	This error message is displayed when a CODESYS address (input) has been selected for an "Analog Output" information object.
Error: Incorrect data type used for "Analog Input" '###1'. Change the defined data type.	This error message is displayed when a CODESYS address (output) has been selected and also entered in the "32-bit" selection field for an "Analog Input" information object.
Error: A maximum of '###1' ETHERNET and serial connections can be configured. Reduce the number of connections.	2 or 4 connections can be defined, depending on the target system. Therefore, reduce the number of configured connection channels to a figure below that for the "Link Layer" item.
Error: Device '###1' does not support the selected COM Port '###2'. Select a different port.	This error message is displayed when a COM port is selected which is not supported by the target system. For example, the COM1 interface cannot be selected for the 750-872 fieldbus controller.

Table 24: Error Messages in the Status Bar

Error message	Description
Error: For analog inputs it is not possible to configure "Float" values for the Level 2 conformity. Change the data type of the object or delete the object.	This error message is displayed when "FLOAT" objects have been defined and the checkbox "Level 2 Conformity" is marked. Remove all objects which use the "FLOAT" type, or change the data type. As an alternative, you can also unmark the "Level 2 Conformity" checkbox. These error messages are only relevant for the information objects "Analog Input" and "Analog Output".
Error: For analog outputs it is not possible to configure "32 Bit" values for the Level 2 conformity. Change the data type of the object or delete the object.	This error message is displayed when "32-bit" objects have been defined and the checkbox "Level 2 Conformity" is marked. Remove all objects which use the "32-bit" type, or change the data type. As an alternative, you can also unmark the "Level 2 Conformity" checkbox. This error message applies only to "Analog Output" information objects.
Error: Multiple references to COM port '###1'. Select a different port.	The identical COM port has been configured for more than one serial interface. Check the serial connections defined in your configuration and ensure that each COM port is used only once.
Error: If the IP filter is enabled, one valid IP address for each connection must be defined at least.	This error message is displayed when the IP filter is activated and not at least one IP address is entered. Deactivate the filter or enter at least one valid IP address. Invalid addresses: 0.0.0.0 / 255.255.255.255
Error: If the master address validation is enabled, a valid IP filter must be set. Or the master address verification must be disabled.	This error message is displayed when the IP filter is deactivated and the address validation is activated.

4.4 Diagnostics Messages via Dialog

Diagnostics messages are displayed in a separate dialog:

Diagnostics Message “Error”

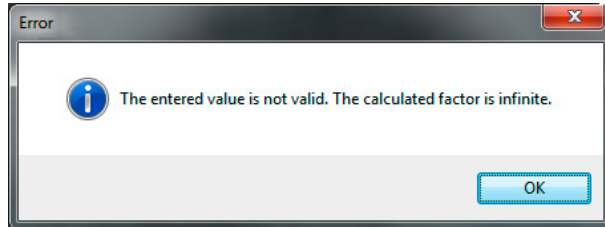


Figure 34: Diagnostics Message “Error”

Factor and offset are calculated as soon as the scaling value is entered. This error message is displayed when no correct factor can be calculated on account of the minimum and maximum values that have been entered. Check the values you have entered for plausibility, or whether the factor has been incorrectly calculated.

Diagnostics Message “Error”

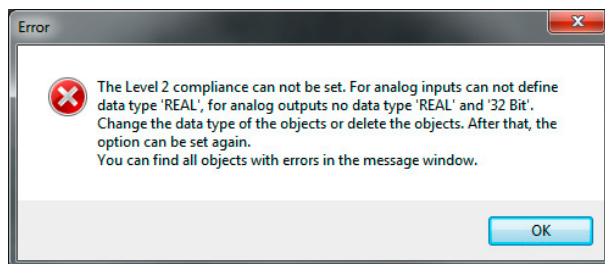


Figure 35: Diagnostics Message “Error”

This message is displayed when “FLOAT” or “32-bit” objects have been defined and the checkbox “Level 2 Conformity” is marked for the “DNP Outstation” item.

Diagnostics Message “Information”

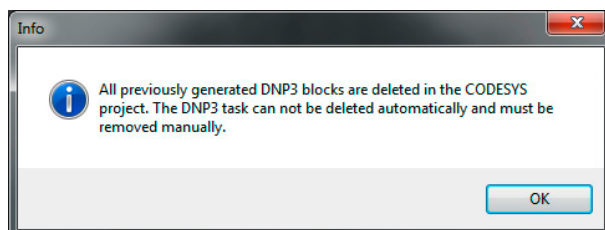


Figure 36: Diagnostics Message “Information”

This message is displayed when the Configurator is closed using the [OK] button and the current project has been deleted.

Diagnostics Message “Warning”

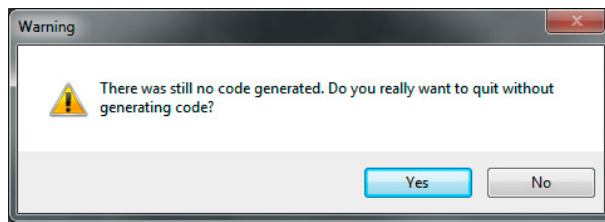


Figure 37: Diagnostics Message “Warning”

This message is displayed when the Configurator is closed using the **[Cancel]** button and an existing project has been modified. You can select whether you wish to cancel all of your changes, or return to the configuration.

5 Appendix

Based on DNP3, a number of information objects are subdivided into individual groups. As a rule, there are then several variations within the individual groups which determine the transmission format. All of the defined information objects and the associated variations are listed in the table below.

Table 25: Information Objects and Associated Variations for DNP3

Group	Description	Variant											
		1	2	3	4	5	6	7	8	9	10	11	12
1	binary input static objects	X	X	-	-	-	-	-	-	-	-	-	-
2	binary input event objects	X	X	X	-	-	-	-	-	-	-	-	-
3	double-bit binary input static objects	⊗	⊗	-	-	-	-	-	-	-	-	-	-
4	double-bit binary input event objects	⊗	⊗	⊗	-	-	-	-	-	-	-	-	-
10	binary output static objects	⊗	X	-	-	-	-	-	-	-	-	-	-
11	binary output event objects	⊗	⊗	-	-	-	-	-	-	-	-	-	-
12	binary output command objects	X	⊗	⊗	-	-	-	-	-	-	-	-	-
13	binary output command event objects	⊗	⊗	-	-	-	-	-	-	-	-	-	-
20	counter static objects	X	X	O	O	X	X	⊗	⊗	-	-	-	-
21	frozen counter static objects	X	X	O	O	X	X	O	O	X	X	O	O
22	counter event objects	X	X	O	O	⊗	⊗	O	O	-	-	-	-
23	frozen counter event objects	⊗	⊗	O	O	⊗	⊗	O	O	-	-	-	-
30	analog input static objects	X	X	X	X	X	⊗	-	-	-	-	-	-
31	frozen analog input static objects	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	-	-	-	-
32	analog input event objects	X	X	X	X	X	⊗	X	⊗	-	-	-	-
33	frozen analog input event objects	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	-	-	-	-
34	analog input deadband objects	⊗	⊗	⊗	-	-	-	-	-	-	-	-	-
40	analog output status objects	X	X	X	⊗	-	-	-	-	-	-	-	-

Table 25: Information Objects and Associated Variations for DNP3

Group	Description	Variant											
		1	2	3	4	5	6	7	8	9	10	11	12
41	analog output command objects	X	X	X	X	-	-	-	-	-	-	-	-
42	analog output event objects	X	X	X	X	X	X	X	-	-	-	-	-
43	analog output command event objects	X	X	X	X	X	X	X	-	-	-	-	-

Table 26: Legend for “Information Objects and Associated Variations for DNP3”

Symbol	Description
X	Object implemented in the WAGO solution
X	Object not implemented in the WAGO solution
O	Object obsolete for DNP3
-	Object not defined for DNP3

6 Glossary

A

Analog Input Group

A group of all “Analog Input” data points

Analog Output Group

A group of all “Analog Output” data points

Application Layer

An application layer in the ISO/OSI model which provides services and methods, for example, for sending or receiving data telegrams.

B

Binary Input Group

A group of all “Digital Input” data points

Binary Output Group

A group of all “Digital Output” data points

Broadcast

Data telegrams are sent simultaneously to all subscribers in the network.

C

Control Direction

Direction of data transmission: from the master station to the substation

D

Data Link Layer

A data link layer in the ISO/OSI model which lies below the transport layer and which is responsible for station addressing and error detection.

DNP3 Device Profile File

A manufacturer-independent, XML-based file format defined in the standard that is used for the exchange of communication parameters, data points and descriptive information.

E**Event**

An event occurs which initiates the generation of an entry in the memory.

Event Data

Denotes a data entry. This data is entered into the → *Event Buffer* based on an event that has occurred. The entry contains event-related data, such as the type and updated value of a data point and the time flag indicating the time that a value changed.

Event Buffer

Buffer (memory) which retains data until it is requested by a master station.

F**Flag**

Boolean variable normally used as a status indicator

FLOAT

Data type for floating point numbers

Fragment

Data packet which corresponds to the size of the input buffer for a device. Here, each fragment contains information about the application layer, such as queries, replies or acknowledgements.

Frame

Data packet exchanged between the link layers of two devices. Every frame contains a connection header, a checksum to check for data integrity and, where applicable, also transport information.

Freeze Request

Command used to “freeze” data points at a defined time

Frozen Counter

Counter value which can be frozen at a defined time by the appropriate command

K

Keep alive

Mechanism used for checking the connection. This mechanism sends telegrams at regular time intervals while communication is inactive on a connection. The connection status is determined when corresponding acknowledgements are received by the opposite station.

M

Master-Station

Telecontrol central station, which sets up a connection to one (or more) substation(s) and requests or receives process data from there or sends command data to the substation(s).

Monitoring Direction

Direction of data transmission: from the substation to the master station.

O

Obsolete Function

Obsolete function or request which should no longer be implemented on account of changed requirements in the standards

Offset

Constant, positive or negative offset for a condition or measured value

P

Point

Data point, normally a digital or analog input or output whose value can be queried and/or written via an address

Point Index

Data point address, which permits unique addressing within a device (where applicable also in combination with other information)

Point Type

Type of data point which can contain different characteristic attributes: for example, binary inputs, analog inputs, counters, binary outputs or analog outputs.

Poll

Querying of data by a master station

Polling

Regular/cyclic querying of a data point or several data points

R**REAL**

Data type for floating point numbers

U**Unsolicited Response**

Telegram at the application layer, sent from a substation to a master station. The master station did not explicitly request this telegram however.

V**Variation**

Data packet transmission format. This defines, for example, whether a data point is to be transmitted with or without flags.

List of Figures

Figure 1: “Target Settings” Dialog.....	16
Figure 2: “New POU” dialog	17
Figure 3: Program Editor “PLC_PRG (PRG-ST)”	17
Figure 4: DNP3 Configurator – Add “Connector Ethernet” Object	18
Figure 5: Required Settings for the “Connector Ethernet” Object.....	19
Figure 6: DNP3 Configurator – Add “Analog Input” Information Object	19
Figure 7: Required Settings for the “[000] Analog Input” Information Object	20
Figure 8: Program Editor “DNP3_GVL”	21
Figure 9: Assigning Values in the “DNP3_GVL” Program Editor.....	21
Figure 10: “DNP Master Station” Dialog.....	22
Figure 11: “Configuration” Tab	23
Figure 12: Assigned Values Are Output	24
Figure 13: DNP3 Configurator Overview	25
Figure 14: “File” Menu	26
Figure 15: “Configuration” Menu	27
Figure 16: “Help” Menu.....	27
Figure 17: DNP3 Workspace Overview.....	28
Figure 18: Setting Options for the “DNP Outstation” Item	29
Figure 19: Setting Options for a “Connector Ethernet” Object.....	31
Figure 20: Setting Options for a “Connector Serial” Object.....	33
Figure 21: Setting Options for the “Group Binary Input” Group Object.....	34
Figure 22: Setting Options for a “Binary Input” Information Object	35
Figure 23: Setting Options for the “Group Binary Output” Group Object	37
Figure 24: Setting Options for a “Binary Output” Information Object.....	38
Figure 25: Setting Options for the “Group Counter” Group Object	40
Figure 26: Setting Options for a “Counter” Information Object.....	41
Figure 27: Setting Options for the “Group Analog Input” Group Object.....	43
Figure 28: Setting Options for an “Analog Input” Information Object	45
Figure 29: Schematic Signal Flow Diagram for “Analog Input” Information Object.....	47
Figure 30: Setting Options for the “Group Analog Output” Group Object	48
Figure 31: Setting Options for an “Analog Output” Information Object.....	50
Figure 32: Schematic Signal Flow Diagram for “Analog Output” Information Object.....	52
Figure 33: Status bar.....	53
Figure 34: Diagnostics Message “Error”.....	55
Figure 35: Diagnostics Message “Error”.....	55
Figure 36: Diagnostics Message “Information”	55
Figure 37: Diagnostics Message “Warning”	56

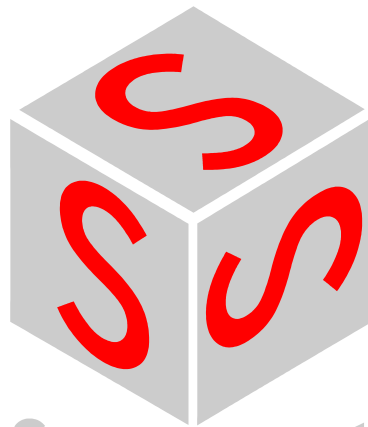
List of Tables

Table 1: Number Notation.....	8
Table 2: Font Conventions	8
Table 3: Abbreviations	13
Table 4: Hardware Requirements.....	15
Table 5: Legend for “DNP3 Configurator Overview”	25
Table 6: “File” Menu.....	26
Table 7: Supported File Formats	26
Table 8: “Configuration” Menu	27
Table 9: “Help” Menu	27
Table 10: Legend for “DNP3 Workspace Overview”	28
Table 11: Setting Options for the “DNP Outstation” Item.....	29
Table 12: Setting Options for a “Connector Ethernet” Object.....	31
Table 13: Setting Options for a “Connector Serial” Object	33
Table 14: Setting Options for the “Group Binary Input” Group Object	35
Table 15: Setting Options for a “Binary Input” Information Object.....	36
Table 16: Setting Options for the “Group Binary Output” Group Object.....	37
Table 17: Setting Options for a “Binary Output” Information Object	38
Table 18: Setting Options for the “Group Counter” Group Object.....	40
Table 19: Setting Options for a “Counter” Information Object	42
Table 20: Setting Options for the “Group Analog Input” Group Object	43
Table 21: Setting Options for an “Analog Input” Information Object.....	45
Table 22: Setting Options for the “Group Analog Output” Group Object.....	48
Table 23: Setting Options for an “Analog Output” Information Object	50
Table 24: Error Messages in the Status Bar	53
Table 25: Information Objects and Associated Variations for DNP3.....	57
Table 26: Legend for “Information Objects and Associated Variations for DNP3”	58

This Page Left Intentionally Blank

User Manual for PLC Programming with

CoDeSys 2.3



S m a r t
S o f t w a r e
S o l u t i o n s

Copyright © 1994, 1997, 1999, 2001, 2002, 2003, 2004, 2005, 2006, 2007 by 3S - Smart Software Solutions GmbH
All rights reserved.

We have gone to great lengths to ensure this documentation is correct and complete. However, since it is not possible to produce an absolutely error-free text, please feel free to send us your hints and suggestions for improving it.

Trademark

Intel is a registered trademark and 80286, 80386, 80486, Pentium are trademarks of Intel Corporation.

Microsoft, MS and MS-DOS are registered trademarks, Windows and Intellisense are trademarks of Microsoft Corporation.

Publisher

3S - Smart Software Solutions GmbH
Memminger Straße 151
D-87439 Kempten
Tel. +49 831 5 40 31 - 0
Fax +49 831 5 40 31 - 50

Last update 17.12.2007

Document Version 5.0, CoDeSys V.2.3.9.0

Content

1	A Brief Introduction to CoDeSys	1-1
1.1	What is CoDeSys.....	1-1
1.2	Overview of CoDeSys Functions.....	1-1
1.3	Overview on the user documentation for CoDeSys.....	1-3
2	What is What in CoDeSys	2-1
2.1	Project Components.....	2-1
2.2	Languages.....	2-9
2.2.1	Instruction List (IL).....	2-9
2.2.2	Structured Text (ST).....	2-11
2.2.3	Sequential Function Chart (SFC).....	2-16
2.2.4	Function Block Diagram (FBD).....	2-21
2.2.5	The Continuous Function Chart Editor (CFC).....	2-21
2.2.6	Ladder Diagram (LD).....	2-22
2.3	Debugging, Online Functions.....	2-23
2.4	The Standard.....	2-26
3	We Write a Little Program	3-1
3.1	Controlling a Traffic Signal Unit.....	3-1
3.2	Visualizing a Traffic Signal Unit.....	3-11
4	The Individual Components	4-1
4.1	The Main Window.....	4-1
4.2	Project Options.....	4-3
4.3	Managing Projects.....	4-22
4.4	Managing Objects in a Project.....	4-53
4.5	General Editing Functions.....	4-60
4.6	General Online Functions.....	4-66
4.7	Window set up.....	4-82
4.8	Help when you need it.....	4-82
5	Editors in CoDeSys	5-1
5.1	This is for all Editors.....	5-1
5.2	Declaration Editor.....	5-3
5.2.1	Working in the Declaration Editor.....	5-3
5.2.2	Declaration Editors in Online Mode.....	5-10
5.2.3	Pragma instructions in the Declaration Editor.....	5-11
5.3	The Text Editors.....	5-19
5.3.1	Working in text editors.....	5-19
5.3.2	The Instruction List Editor.....	5-23
5.3.3	The Editor for Structured Text.....	5-24
5.4	The Graphic Editors.....	5-24
5.4.1	Working in graphic editors.....	5-24
5.4.2	The Function Block Diagram Editor.....	5-28
5.4.3	The Ladder Editor.....	5-33
5.4.4	The Sequential Function Chart Editor.....	5-39

5.4.5	The Continuous Function Chart Editor (CFC).....	5-47
6	The Ressources	6-1
6.1	Overview of the Ressources	6-1
6.2	Global Variables, Variable Configuration, Document Frame	6-2
6.2.1	Global Variables.....	6-2
6.2.2	Variable Configuration.....	6-6
6.2.3	Document Frame	6-7
6.3	Alarm Configuration	6-8
6.3.1	Overview	6-8
6.3.2	General information on alarms, Terms.....	6-9
6.3.3	Alarm classes.....	6-10
6.3.4	Alarm groups.....	6-13
6.3.5	Alarm saving	6-15
6.3.6	'Extras' Menu: Settings.....	6-16
6.4	Library Manager	6-16
6.5	Log	6-18
6.6	PLC Configuration.....	6-20
6.6.1	Overview	6-20
6.6.2	Working in the PLC Configuration.....	6-21
6.6.3	General Settings in the PLC Configuration	6-23
6.6.4	Custom specific parameter dialog.....	6-24
6.6.5	Configuration of an I/O Module	6-25
6.6.6	Configuration of a Channel	6-28
6.6.7	Configuration of Profibus Modules	6-29
6.6.8	Configuration of CANopen modules.....	6-36
6.6.9	Configuration of a CanDevice (CANopen Slave)	6-41
6.6.10	Configuration of DeviceNet Modules.....	6-45
6.6.11	PLC Configuration in Online Mode.....	6-49
6.6.12	Hardware scan/State/Diagnosis information from the PLC	6-50
6.7	Task Configuration	6-51
6.7.1	Overview	6-51
6.7.2	Working in the Task Configuration	6-52
6.7.3	System Events	6-55
6.7.4	Task Configuration in Online Mode.....	6-55
6.8	Watch and Receipt Manager.....	6-57
6.8.1	Overview	6-57
6.8.2	Watch and Receipt Manager in the Offline Mode	6-58
6.8.3	Watch and Receipt Manager in the Online Mode	6-59
6.9	The Sampling Trace.....	6-60
6.9.1	Overview and Configuration.....	6-60
6.9.2	Generating a Trace Sampling	6-62
6.9.3	Looking at the Sampling Trace	6-63
6.9.4	'Extras' 'Save trace values'.....	6-65
6.9.5	'Extras' 'External Trace Configurations'.....	6-66
6.10	Workspace	6-66
6.11	Parameter Manager	6-67
6.11.1	Overview, Activating.....	6-68
6.11.2	The Parameter Manager Editor, Overview.....	6-68
6.11.3	Parameter List Types and Attributes	6-69
6.11.4	Managing parameter lists.....	6-71
6.11.5	Editing parameter lists.....	6-73
6.11.6	Parameter Manager in Online Mode	6-74
6.11.7	Export / Import of parameter lists	6-75

6.12	Target Settings.....	6-75
6.13	The PLC-Browser	6-76
6.13.1	General remarks concerning PLC-Browser operation	6-76
6.13.2	Command entry in the PLC-Browser	6-77
6.13.3	Use of macros during the command entry in PLC-Browser.....	6-79
6.13.4	Further PLC-Browser options	6-79
6.14	Tools	6-80
6.14.1	Properties of available Tool Shortcuts (Object Properties).....	6-80
6.14.2	Managing Tool Shortcuts.....	6-83
6.14.3	Frequently asked questions on Tools	6-84
7	<u>ENI</u>	7-1
7.1.1	What is ENI	7-1
7.1.2	Preconditions for Working with an ENI project data base.....	7-1
7.1.3	Working with the ENI project data base in CoDeSys.....	7-2
7.1.4	Object categories concerning the project data base.....	7-2
8	<u>DDE Interface</u>	8-1
8.1	DDE interface of the CoDeSys programming system.....	8-1
8.2	DDE communication with the GatewayDDE Server.....	8-2
9	<u>The License Management in CoDeSys</u>	9-1
9.1	The License Manager	9-1
9.1.1	Creating a licensed library in CoDeSys	9-1
10	<u>APPENDIX</u>	10-1
Appendix A:	<u>IEC Operators and additional norm extending functions</u>	10-1
10.1	Arithmetic Operators.....	10-1
10.2	Bitstring Operators.....	10-4
10.3	Bit-Shift Operators.....	10-6
10.4	Selection Operators	10-8
10.5	Comparison Operators.....	10-11
10.6	Address Operators.....	10-13
10.7	Calling Operators.....	10-14
10.8	Type Conversions.....	10-14
10.9	Numeric Operators.....	10-20
10.10	Initialization Operator.....	10-24
Appendix B:	<u>Operands in CoDeSys</u>	10-25
10.11	Constants.....	10-25
10.12	Variables	10-27
10.13	Addresses	10-29
10.14	Functions	10-30
Appendix C:	<u>Data types in CoDeSys</u>	10-31
10.15	Standard data types.....	10-31
10.16	Defined data types.....	10-33
Appendix D:	<u>The CoDeSys Libraries</u>	10-39
10.17	The Standard.lib library.....	10-39
10.17.1	String functions.....	10-39

10.17.2	Bistable Function Blocks.....	10-42
10.17.3	Trigger.....	10-44
10.17.4	Counter... ..	10-45
10.17.5	Timer... ..	10-47
10.18	The Util.lib library	10-50
10.18.1	BCD Conversion	10-50
10.18.2	Bit-/Byte Functions	10-50
10.18.3	Mathematic Auxiliary Functions	10-51
10.18.4	Controllers.....	10-54
10.18.5	Signal Generators... ..	10-57
10.18.6	Function Manipulators.....	10-59
10.18.7	Analog Value Processing.....	10-60
10.19	The AnalyzationNew.lib library.....	10-61
10.20	CoDeSys System Libraries	10-62
Appendix E: Operators and Library Modules Overview		10-63
10.21	Operators in CoDeSys	10-63
10.22	Elements of the Standard.lib:	10-66
10.23	Elements of the Util.lib	10-66
Appendix F: Command Line-/Command File		10-69
10.24	Command Line Commands	10-69
10.25	Command File (cmdfile) Commands.....	10-70
Appendix G: Siemens Import		10-79
10.26	Import from a SEQ Symbol File	10-79
10.27	Import from a S5 Project File	10-80
10.28	Converting S5 to IEC 61131-3	10-80
Appendix H: Target Settings in Detail		10-85
10.29	Settings in Category Target Platform	10-85
10.29.1	Target system Intel 386 compatible, Category Target Platform.....	10-85
10.29.2	Target system Motorola 68K, Category Target Platform.....	10-86
10.29.3	Target system Infineon C16x, Category Target Platform.....	10-87
10.29.4	Target systems Intel StrongARM und Power PC, Category Target Platform.....	10-88
10.29.5	Target system MIPS, Category Target Platform	10-89
10.29.6	Target system 'Hitachi SH', Category Target Platform.....	10-90
10.29.7	Target system '8051 compatible', Category Target Platform	10-91
10.29.8	Target System 'TriCore', Categorie Target platform	10-91
10.30	Target Settings for Category Memory Layout	10-92
10.31	Target Settings in Category General.....	10-94
10.32	Target Settings in Category Networkfunctionality	10-96
10.33	Target Settings in Category Visualization	10-97
Appendix I: Use of Keyboard		10-99
10.34	Use of Keyboard	10-99
10.35	Key Combinations	10-99
Appendix J: Recommendations on the naming of identifiers		10-103
10.36	Naming of identifiers	10-103
10.37	Identifiers for variables (variable names)	10-103
10.38	Identifiers for user-defined data types (DUT)	10-105

Table Of Content

10.39	Identifiers for functions, function blocks, programs (POU).....	10-105
10.40	Identifiers for Visualizations	10-106
Appendix K:	<u>Compiler Errors and Warnings</u>	<u>10-107</u>
10.41	Warnings.....	10-107
10.42	Errors	10-113
11	<u>Index</u>	<u>CL</u>

Table Of Content

1 A Brief Introduction to CoDeSys

1.1 What is CoDeSys

CoDeSys is a complete development environment for your PLC. (**CoDeSys** stands for Controlled Development System).

CoDeSys puts a simple approach to the powerful IEC language at the disposal of the PLC programmer. Use of the editors and debugging functions is based upon the proven development program environments of advanced programming languages (such as Visual C++).

1.2 Overview of CoDeSys Functions...

How is a project structured?

A project is put into a file named after the project. The first POU (Program Organization Unit) created in a new project will automatically be named PLC_PRG. The process begins here (in compliance with the main function in a C program), and other POUs can be accessed from the same point (programs, function blocks and functions).

Once you have defined a Task Configuration, it is no longer necessary to create a program named PLC_PRG. You will find more about this in the Task Configuration chapter.

There are different kinds of objects in a project: POUs, data types, display elements (visualizations) and resources.

The Object Organizer contains a list of all the objects in your project.

How do I set up my project?

First you should configure your PLC in order to check the accuracy of the addresses used in the project.

Then you can create the POUs needed to solve your problem.

Now you can program the POUs you need in the desired languages.

Once the programming is complete, you can compile the project and remove errors should there be any.

How can I test my project?

Once all errors have been removed, activate the **simulation**, log in to the simulated PLC and "load" your project in the PLC. Now you are in Online mode.

Now open the window with your **PLC Configuration** and test your project for correct sequence. To do this, enter input variables manually and observe whether outputs are as expected. You can also observe the value sequence of the local variables in the POUs. In the **Watch and Receipt Manager** you can configure data records whose values you wish to examine.

Debugging

In case of a programming error you can set breakpoints. If the process stops at such a breakpoint, you can examine the values of all project variables at this point in time. By working through sequentially (single step) you can check the logical correctness of your program.

Additional Online Functions

Further debugging functions:

You can **set** program variables and inputs and outputs at certain values.

You can use the **flow control** to check which program lines have been run.

A **Log** records operations, user actions and internal processes during an online session in a chronological order.

If activated in the target settings the **Sampling Trace** allows you to trace and display the actual course of variables over an extended period of time.

Also a target specific function is the **PLC Browser** which can serve to request certain information from the PLC.

Once the project has been set up and tested, it can be loaded down to the hardware and tested as well. The same online functions as you used with the simulation will be available.

Additional CoDeSys Features

The entire project can be documented or exported to a text file at any time.

For **communication** purposes CoDeSys has a symbolic interface and a DDE interface. A Gateway Server plus OPC Server and DDE Server are components of the **CoDeSys**-standard installation packet.

Using the appropriate **target settings**, which can be loaded with the aid of a target file (Target Support Package) allows to load the same CoDeSys project to various target systems.

Network global variables and a **Parameter Manager** might be available, if activated by the current target settings, for data exchange within a network of controllers.

ENI: The 'Engineering Interface' can be used to access any desired source code management program via the ENI Server, which is running as an independent process. CoDeSys POU's and compile files can be filed in that data base and are by that accessible also by other clients of the ENI Server. This allows multi user operation during the work on a CoDeSys project, it provides a common data pool for different tools besides CoDeSys and it makes possible a version management.

Tools: This functionality also is target dependent and allows to start target-specific executable files in a CoDeSys project. Besides that files can be defined, which should be loaded to the controller. Connections to external tools can be pre-defined in the target file or/and inserted in the project Resource tree.

A **CoDeSys visualization** can be processed target specifically to be available as **Web-Visualization** and/or **Target-Visualization**. This allows to run and view the visualization via Internet or on a PLC-monitor.

1.3 Overview on the user documentation for CoDeSys

Module	Docu Contents	Name of File
CoDeSys Programming System	On hand manual and online help via help menu in the programming system First steps with the CoDeSys Programming system (sample)	CoDeSys_V23_E.pdf First Steps with CoDeSys V23.pdf
Gateway Server	Concept, installation and User Interface; Online Help for the User Interface via Gateway menu (can be opened by a mouse-click on the gateway symbol in the system tray)	Gateway Manual.pdf
OPC Server	OPC-Server V2.0, Installation and Use	OPC_20_How_to_use_E.pdf
CoDeSys Visualization	Manual for the CoDeSys Visualization incl. CoDeSys HMI, Target- and Web-Visualization	CoDeSys_Visu_V23_E.pdf
SoftMotion	How to use, description of the main SoftMotion library modules	CoDeSys_SoftMotion_V23_E.pdf
Bibliotheken	Standard.lib and Util.lib are described in the on hand manual. For each of the CoDeSys System Libraries there is a document <library name>.pdf SoftMotion libraries: see SoftMotion-documentation.	<SysLib-Name>.pdf CoDeSys_V23_E.pdf
ENI Server	Installation and configuration of the ENI Servers concerning the source control of a CoDeSys project in an external data base. Configuration of ENI in CoDeSys: see the on hand manual. ENI Admin, ENI Control and ENI Explorer: see the referring online help.	EniServerQuickstart_E.pdf CoDeSys_V23_E.pdf

2 What is What in CoDeSys

2.1 Project Components...

Project

A project contains all of the objects in a PLC program. A project is saved in a file named after the project. The following objects are included in a project:

POUs (Program Organization Units), data types, visualizations, resources, and libraries.

POU (Program Organization Unit)

Functions, function blocks, and programs are POUs which can be supplemented by actions.

Each POU consists of a declaration part and a body. The body is written in one of the IEC programming languages which include IL, ST, SFC, FBD, LD or CFC.

CoDeSys supports all IEC standard POUs. If you want to use these POUs in your project, you must include the library standard.lib in your project.

POUs can call up other POUs. However, recursions are not allowed.

Function

A function is a POU, which yields exactly one data element (which can consist of several elements, such as fields or structures) when it is processed, and whose call in textual languages can occur as an operator in expressions.

When declaring a function do not forget that the function must receive a type. This means, after the function name, you must enter a colon followed by a type. Regard Appendix J: Recommendations on the naming.

A correct function declaration can look like this example:

```
FUNCTION Fct: INT
```

In addition, a result must be assigned to the function. That means that function name is used as an output variable.

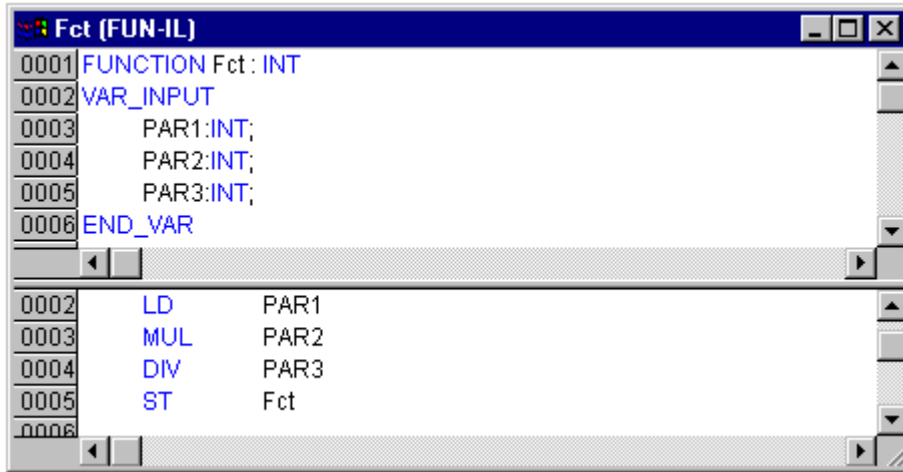
A function declaration begins with the keyword FUNCTION.

In IL a function call only can be positioned within actions of a step or within a transition.

In ST a function call can be used as operand in an expression.

Example in IL of a function that takes three input variables and returns the product of the first two divided by the third:

Example of a function in IL



Calling a function:

The call of a function in ST can appear as an operand in expressions.

In SFC a function call can only take place within a step or a transition.

Note: CoDeSys allows using global variables within a function. This intentionally deviates from the IEC61131-3 standard, which prescribes that return value of a function only will be modified by the input parameters. Thus the difference between functions and programs is just, that functions return only one return value and that their parameters and return values are handled over the stack.

Examples for calling up a function.

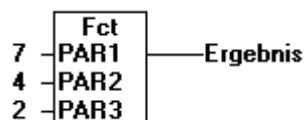
in IL:

```
LD 7
Fct 2,4
ST Result
```

in ST:

```
Result := Fct(7, 2, 4);
```

in FBD:



Attention: If a local variable is declared as RETAIN in a function, this is without any effect ! The variable will not be written to the Retain area !

Note: The following check function names are reserved for the described usage:

- If you define a function in your project with the name **CheckBounds**, you can use it to check range overflows in your project! The name of the function is defined and may have only this identifier. For further description please see chapter 10.1, Arithmetic Operators, DIV.
- If you define functions in your project with the names **CheckDivByte**, **CheckDivWord**, **CheckDivDWord** resp. **CheckDivReal**, you can use them to check the value of the divisor if you use the operator DIV, for example to avoid a division by 0.
- If you define functions with the names **CheckRangeSigned** and **CheckRangeUnsigned**, then range exceeding of variables declared with subrange types (see chapter 10.16, Data types) can be intercepted.

Function Block

A function block is a POU which provides one or more values during the procedure. As opposed to a function, a function block provides no return value.

A function block declaration begins with the keyword `FUNCTION_BLOCK`. Regard Appendix J: Appendix J: Recommendations on the naming.

Reproductions or instances (copies) of a function block can be created.

Example of a function block in IL

```

0001 FUNCTION_BLOCK FBD
0002 VAR_INPUT
0003   PAR1:INT;
0004   PAR2:INT;
0005 END_VAR
0006 VAR_OUTPUT
0007   MULERG:INT;
0008   VERGL:BOOL;
0009 END_VAR
0010 VAR
0011 END_VAR

0001 LD   PAR1
0002 MUL  PAR2
0003 ST   MULERG
0004
0005 LD   PAR1
0006 EQ   PAR2
0007 ST   VERGL

```

Example in IL of a function block with two input variables and two output variables. One output is the product of the two inputs, the other a comparison for equality:

Function Block Instances

Reproductions or *instances* (copies) of a function block can be created.

Each instance possesses its own identifier (the Instance name), and a data structure which contains its inputs, outputs, and internal variables. Instances are declared locally or globally as variables, whereas the name of the function block is indicated as the type of an identifier. Regard Appendix J: Appendix J: Recommendations on the naming.

Example of an instance with the name `INSTANCE` of the `FUB` function block:

```
fubinstance: FUB;
```

Function blocks are always called through the instances described above.

Only the input and output parameters can be accessed from outside of a function block instance, not its internal variables.

Example for accessing an input variable

The function block `FB` has an input variable `in1` of the type `INT`.

```

PROGRAM prog
VAR
fbinst1:fb;
END_VAR

LD 17
ST fbinst1.iIn1
CAL fbinst1

END_PROGRAM

```


The declaration parts of function blocks and programs can contain instance declarations. Instance declarations are not permitted in functions.

Access to a function block instance is limited to the POU in which it was declared unless it was declared globally.

The instance name of a function block instance can be used as the input for a function or a function block.

Note:	All values are retained after processing a function block until the next it is processed. Therefore, function block calls with the same arguments do not always return the same output values!
Note:	If there at least one of the function block variables is a retain variable, the total instance is stored in the retain area.

Calling a function block

The input and output variables of a function block can be accessed from another POU by setting up an instance of the function block and specifying the desired variable using the following syntax:

<Instance name>.<Variable name>

Assigning parameters at call:

If you would like to set input and/or output parameters when you call the function block, you can do this in the text languages IL and ST by assigning values to the parameters after the instance name of the function block in parentheses (for input parameters this assignment takes place using "=" just as with the initialization of variables at the declaration position, for output parameters ">=" is to be used).

If the instance is inserted via input assistant (<F2>) with option **With arguments** in the implementation window of a ST or IL POU, it will be displayed automatically according to this syntax with all of its parameters. But you not necessarily must assign these parameters.

Example:

FBINST is a local variable of type of a function block, which contains the input variable xx and the output variable yy. When FBINST is inserted in a ST program via input assistant, the call will be displayed as follows: FBINST1(xx:= , yy=>);

InOutVariables at call:

Please regard, that the InOutVariables (VAR_IN_OUT) of a function block are handed over as pointers. For this reason in a call of a function block no constants can be assigned to VAR_IN_OUTs and there is no read or write access from outside to them.

Example

Calling a VAR_IN_OUT variable iInOut1 of function block fubo in a ST module:

```
VAR
  fuboinst:fubo;
  iVar1:int;
END_VAR
iVar1:=2;
fuboinst(iInOut1:=iVar1);
```

not allowed in this case: "fuboinst(iInOut1:=2);" or "fuboinst.iInOut1:=2;"

Examples for calling function block FUB:

Function block FUB, see above, chapter 'Function Block'

The multiplication result is saved in the variable ERG, and the result of the comparison is saved in QUAD. An instance of FUB with the name INSTANCE is declared.

This is how the instance of a function block is called in IL :

```

AWLCall (PRG-IL)
0001 PROGRAM AWLCall
0002 VAR
0003     QUAD: BOOL;
0004     INSTANCE: FUB;
0005     ERG: INT:=0;
0006 END_VAR

0001     CAL     INSTANCE(PAR1:=5,PAR2:=5)
0002     LD     INSTANCE.VERGL
0003     ST     QUAD
0004     LD     INSTANCE.MULERG
0005     ST     ERG
0006
    
```

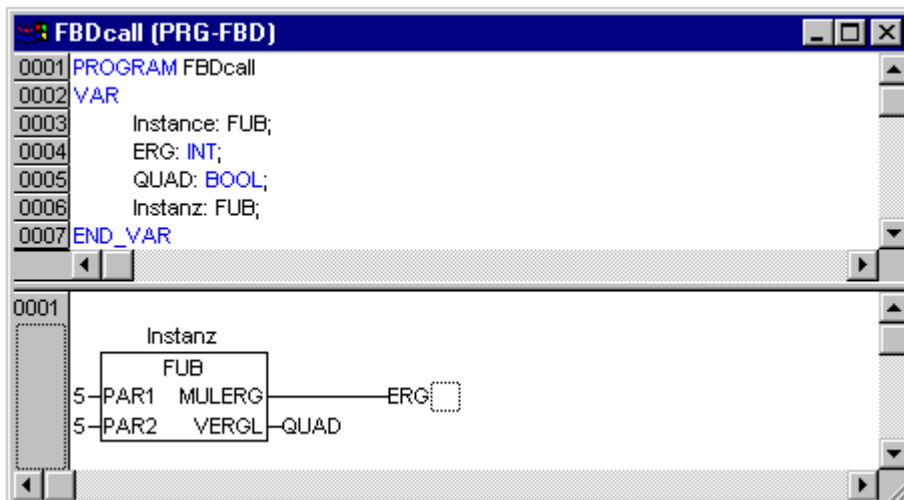
This is how the instance of a function block is called in ST (the declaration part the same as with IL)

```

STcall (PRG-ST)
0001 PROGRAM STcall

0001 INSTANZ(PAR1:=5,PAR2:=5);
0002 QUAD:=INSTANZ.VERGL;
0003 ERG:=INSTANZ.MULERG;
    
```

This is how the instance of a function block is called in FBD (the declaration part the same as with IL)



In SFC function block calls can only take place in steps.

Program

A program is a POU which returns several values during operation. Programs are recognized globally throughout the project. All values are retained from the last time the program was run until the next.

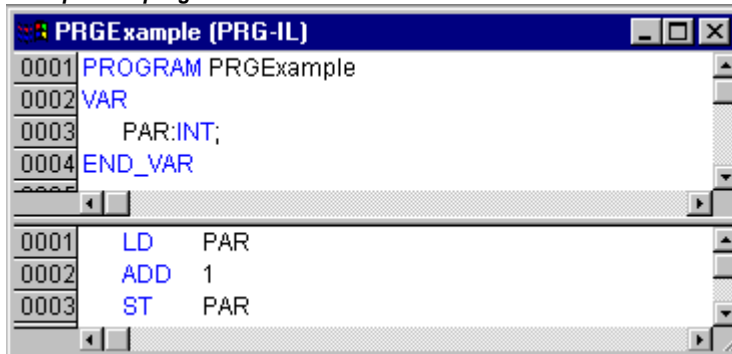
Programs can be called. A program call in a function is not allowed. There are also no instances of programs.

If a POU calls a program, and if thereby values of the program are changed, then these changes are retained the next time the program is called, even if the program has been called from within another POU.

This is different from calling a function block. There only the values in the given instance of a function block are changed.

These changes therefore play a role only when the same instance is called.

Example of a program



A program declaration begins with the keyword PROGRAM and ends with END_PROGRAM. Regard Appendix J: Appendix J: Recommendations on the naming.

If you would like to set input and/or output parameters when you call the program, you can do this in the text languages IL and ST by assigning values to the parameters after the program name in parentheses (for input parameters this assignment takes place using "!=" just as with the initialization of variables at the declaration position, for output parameters "=>" is to be used).

If the program is inserted via input assistant (<F2>) with option **With arguments** in the implementation window of a ST or IL POU, it will be displayed automatically according to this syntax with all of its parameters. But you not necessarily must assign these parameters.

Examples for program calls:

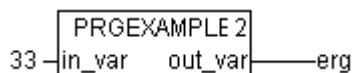
In IL:

```
CAL PRGexample2
LD PRGexample2.out_var
ST ERG
or with assigning the parameters (input assistant "With arguments", see above):
CAL PRGexample2(in_var:=33, out_var=>erg )
```

In ST:

```
PRGexample2;
Erg := PRGexample2.out_var;
or with assigning the parameters (input assistant "With arguments", see above):
PRGexample2(in_var:=33, out_var=>erg );
```

In FBD:



Example for a possible call sequence for PLC_PRG:

See the program PRGexample shown in the picture at top of this chapter:

```
LD 0
ST PRGexample.PAR (*Default setting for PAR is 0*)
CAL IL call (*ERG in IL call results in 1*)
CAL ST call (*ERG in ST call results in 2*)
CAL FBD call (*ERG in FBD call results in 3*)
```

If the variable PAR from the program PRGexample is initialized by a main program with 0, and then one after the other programs are called with above named program calls, then the ERG result in the programs will have the values 1, 2, and 3. If one exchanges the sequence of the calls, then the values of the given result parameters also change in a corresponding fashion.

PLC_PRG

The PLC_PRG is a special predefined POU. Each project must contain this special program. This POU is called exactly once per control cycle.

The first time the **'Project' 'Object Add'** command is used after a new project has been created, the default entry in the POU dialog box will be a POU named PLC_PRG of the program type. You should not change this default setting!

If tasks have been defined, then the project may not contain any PLC_PRG, since in this case the procedure sequence depends upon the task assignment.

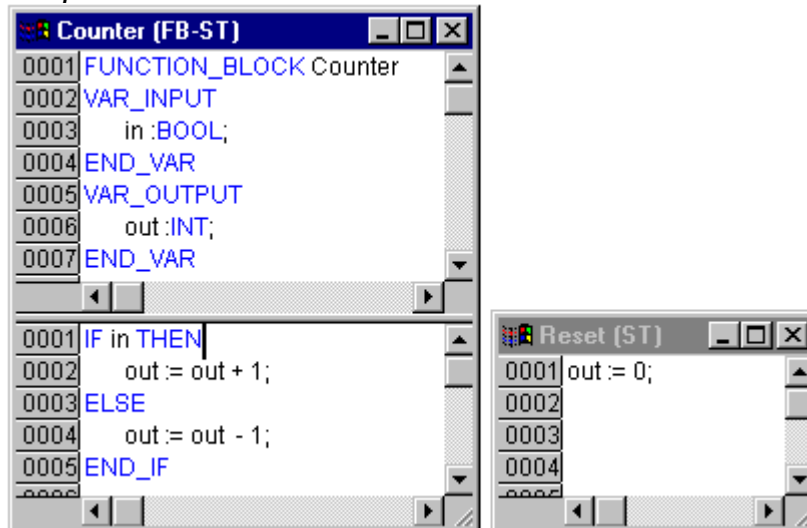
Attention: Do not delete or rename the POU PLC_PRG (assuming you are not using a Task Configuration)! PLC_PRG is generally the main program in a single task program.

Action

Actions can be defined and assigned to function blocks and programmes ('Project' 'Add action'). The action represents a further implementation which can be entirely created in another language as the "normal" implementation. Each action is given a name.

An action works with the data from the function block or programme which it belongs to. The action uses the same input/output variables and local variables as the "normal" implementation uses.

Example for an action of a function block



In the example given, calling up the function block Counter increases or decreases the output variable "out", depending on the value of the input variable "in". Calling up the action Reset of the function block sets the output variable to zero. The same variable "out" is written in both cases.

Calling an action:

An action is called up with <Program_name>.<Action_name> or <Instance_name>.<Action_name>. Regard the notation in FBD (see example below) ! If it is required to call up the action within its own block, one just uses the name of the action in the text editors and in the graphic form the function block call up without instance information.

Examples for calls of the above described action from another POU:

Declaration for all examples:

```
PROGRAM PLC_PRG
VAR
  Inst : Counter;
END_VAR
```

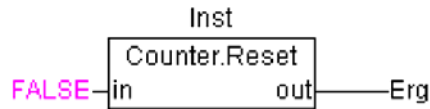
Call of action 'Reset' in another POU, which is programmed in IL:

```
CAL Inst.Reset(In := FALSE)
LD Inst.out
ST ERG
```

Call of action 'Reset' in another POU, which is programmed in **ST**:

```
Inst.Reset(In := FALSE);
Erg := Inst.out;
```

Call of action 'Reset' in another POU, which is programmed in **FBD**:



Note:	Actions play an important role in blocks in sequential function charts, see Sequential Function Chart. The IEC standard does not recognise actions other than actions of the sequential function chart.
--------------	---

Resources

You need the resources for configuring and organizing your project and for tracing variable values:

Global Variables which can be used throughout the project or network

Library manager for adding libraries to the project

Log for recording the actions during an online session

Alarm Configuration for the configuration of the alarm handling in the project

PLC Configuration for configuring your hardware

Task Configuration for guiding your program through tasks

Watch and Receipt Manager for displaying variable values and setting default variable values

Target Settings for selection and if necessary final configuration of the target system

Workspace as an image of the project options

Depending on the target system and on the target settings made in CoDeSys the following resources also might be available in your project:

Sampling Trace for graphic display of variable values

Parameter Manager for data exchange with other controllers in a network

PLC-Browser as controller monitor

Tools – availability depending on target – for calling external tool programs from within CoDeSys

SoftMotion functionality (due to license) with CNC-Editor (CNC program list) and CAM-Editor

Libraries

You can include in your project a series of libraries whose POU's, data types, and global variables you can use just like user-defined variables. The libraries standard.lib and util.lib are standard parts of the program and are always at your disposal. See chapter 6.4 'Library Manager'.

Data types

Along with the standard data types the user can define his own data types. Structures, enumeration types and references can be created.

See Appendix C: 'Data Types'.

Visualization

CoDeSys provides visualizations so that you can display your project variables. You can plot geometric elements off-line with the help of the visualization. In Online mode, these can then change their form/color/text output in response to specified variable values.

A visualization can be used as a pure operating interface for a PLC with CoDeSys HMI or as a Web-Visualization or Target-Visualization running via Internet resp. directly on the PLC.

See for more details in the user manual for the CoDeSys Visualization.

2.2 Languages...

CoDeSys supports all languages described by the standard IEC-61131:

Textual Languages:

Instruction List (IL)

Structured Text (ST)

Grafic Languages:

Sequential Function Chart (SFC)

Function Block Diagram (FBD)

Ladder Diagram (LD)

Additionally there is available - basing on the Function Block Diagram - the grafic Continuous Function Chart Editor (CFC).

2.2.1 Instruction List (IL)...

An instruction list (IL) consists of a series of instructions. Each instruction begins in a new line and contains an operator and, depending on the type of operation, one or more operands separated by commas.

In front of an instruction there can be an identification *mark* (label) followed by a colon (:).

A comment must be the last element in a line. Empty lines can be inserted between instructions.

Example:

```
LD 17
ST lint (* Kommentar *)
GE 5
JMPC next
LD idword
EQ istruct.sdword
STN test
next:
```

Modifiers and operators in IL

In the IL language the following operators and modifiers can be used.

Modifiers:

- C with JMP, CAL, RET: The instruction is only then executed if the result of the preceding expression is TRUE.
- N with JMPC, CALC, RETC: The instruction is only then executed if the result of the preceding expression is FALSE.
- N otherwise: Negation of the operand (not of the accumulator)

Below you find a table of all operators in IL with their possible modifiers and the relevant meaning:

Operator Modifiers Meaning

LD	N	Make current result equal to the operand
ST	N	Save current result at the position of the operand
S		Then put the Boolean operand exactly at TRUE if the current result is TRUE

R		Then put the Boolean operand exactly at FALSE if the current result is TRUE
AND	N,(Bitwise AND
OR	N,(Bitwise OR
XOR	N,(Bitwise exclusive OR
ADD	(Addition
SUB	(Subtraction
MUL	(Multiplication
DIV	(Division
GT	(>
GE	(>=
EQ	(=
NE	(<>
LE	(<=
LT	(<
JMP	CN	Jump to the label
CAL	CN	Call program or function block or
RET	CN	Leave POU and return to caller.
)		Evaluate deferred operation

[Click here to get a listing of all IEC operators.](#)

Example of an IL program while using some modifiers:

```
LD TRUE (*load TRUE in the accumulator*)
ANDN BOOL1 (*execute AND with the negated value of the BOOL1 variable*)
JMPC mark (*if the result was TRUE, then jump to the label "mark"*)

LDN BOOL2 (*save the negated value of *)
ST ERG (*BOOL2 in ERG*)

label:
LD BOOL2 (*save the value of *)
ST ERG *BOOL2 in ERG*)
```

It is also possible in IL to put **parentheses after an operation**. The value of the parenthesis is then considered as an operand.

For example:

```
LD 2
MUL 2
ADD 3
Erg
```

Here is the value of Erg 7. However, if one puts parentheses:

```

LD 2
MUL (2
ADD 3
)
ST Erg

```

the resulting value for Erg is 10, the operation MUL is only then evaluated if you come to ")"; as operand for MUL 5 is then calculated.

2.2.2 Structured Text (ST)...

The Structured Text consists of a series of instructions which, as determined in high level languages, ("IF..THEN..ELSE") or in loops (WHILE..DO) can be executed.

Example:

```

IF value < 7 THEN
  WHILE value < 8 DO
    value:=value+1;
  END WHILE;
END_IF;

```

Expressions

An *expression* is a construction which returns a value after its evaluation.

Expressions are composed of operators and operands. An operand can be a constant, a variable, a function call, or another expression.

Valuation of expressions

The evaluation of expression takes place by means of processing the operators according to certain *binding rules*. The operator with the strongest binding is processed first, then the operator with the next strongest binding, etc., until all operators have been processed.

Operators with equal binding strength are processed from left to right.

Below you find a table of the ST operators in the order of their binding strength:

Operation	Symbol	Binding strength
Put in parentheses	(expression)	Strongest binding
Function call	Function name (parameter list)	
Exponentiation	EXPT	
Negate	-	
Building of complements	NOT	
Multiply	*	
Divide	/	
Modulo	MOD	
Add	+	
Subtract	-	
Compare	<,>,<=,>=	
Equal to	=	
Not equal to	<>	
Boolean AND	AND	
Boolean XOR	XOR	

Boolean OR OR Weakest binding

There are the following instructions in ST, arranged in a table together with example:

Instruction type	Example
Assignment	A:=B; CV := CV + 1; C:=SIN(X);
Calling a function block and use of the FB output	CMD_TMR(IN := %IX5, PT := 300); A:=CMD_TMR.Q
RETURN	RETURN;
IF	D:=B*B; IF D<0.0 THEN C:=A; ELSIF D=0.0 THEN C:=B; ELSE C:=D; END_IF;
CASE	CASE INT1 OF 1: BOOL1 := TRUE; 2: BOOL2 := TRUE; ELSE BOOL1 := FALSE; BOOL2 := FALSE; END_CASE;
FOR	J:=101; FOR I:=1 TO 100 BY 2 DO IF ARR[I] = 70 THEN J:=I; EXIT; END_IF; END_FOR;
WHILE	J:=1; WHILE J<= 100 AND ARR[J] <> 70 DO J:=J+2; END_WHILE;
REPEAT	J:=-1; REPEAT J:=J+2; UNTIL J= 101 OR ARR[J] = 70 END_REPEAT;
EXIT	EXIT;
Empty instruction	;

Assignment operator

On the left side of an assignment there is an operand (variable, address) to which is assigned the value of the expression on the right side with the assignment operator :=

Example:

```
Var1 := Var2 * 10;
```

After completion of this line Var1 has the tenfold value of Var2.

Calling function blocks in ST

A function block is called in ST by writing the name of the instance of the function block and then assigning the values of the parameters in parentheses. In the following example a timer is called with assignments for the parameters IN and PT. Then the result variable Q is assigned to the variable A.

The result variable, as in IL, is addressed with the name of the function block, a following point, and the name of the variable:

```
CMD_TMR(IN := %IX5, PT := 300);
A:=CMD_TMR.Q
```

RETURN instruction

The RETURN instruction can be used to leave a POU, for example depending on a condition

IF instruction

With the IF instruction you can check a condition and, depending upon this condition, execute instructions.

Syntax:

```
IF <Boolean_expression1> THEN
<IF_instructions>
{ELSIF <Boolean_expression2> THEN
<ELSIF_instructions1>
.
.
ELSIF <Boolean_expression n> THEN
<ELSIF_instructions n-1>
ELSE
<ELSE_instructions>}
END_IF;
```

The part in braces {} is optional.

If the <Boolean_expression1> returns TRUE, then only the <IF_Instructions> are executed and none of the other instructions.

Otherwise the Boolean expressions, beginning with <Boolean_expression2>, are evaluated one after the other until one of the expressions returns TRUE. Then only those instructions after this Boolean expression and before the next ELSE or ELSIF are evaluated.

If none of the Boolean expressions produce TRUE, then only the <ELSE_instructions> are evaluated.

Example:

```
IF temp<17
THEN heating_on := TRUE;
ELSE heating_on := FALSE;
END_IF;
```

Here the heating is turned on when the temperature sinks below 17 degrees. Otherwise it remains off.

CASE instruction

With the CASE instructions one can combine several conditioned instructions with the same condition variable in one construct.

Syntax:

```
CASE <Var1> OF
<Value1>: <Instruction 1>
<Value2>: <Instruction 2>
<Value3, Value4, Value5>: <Instruction 3>
<Value6 .. Value10>: <Instruction 4>
...
<Value n>: <Instruction n>
ELSE <ELSE instruction>
END_CASE;
```

A CASE instruction is processed according to the following model:

If the variable in <Var1> has the value <Value i>, then the instruction <Instruction i> is executed.

If <Var 1> has none of the indicated values, then the <ELSE Instruction> is executed.

□ If the same instruction is to be executed for several values of the variables, then one can write these values one after the other separated by commas, and thus condition the common execution.

If the same instruction is to be executed for a value range of a variable, one can write the initial value and the end value separated by two dots one after the other. So you can condition the common condition.

Example:

```
CASE INT1 OF
1, 5: BOOL1 := TRUE;
      BOOL3 := FALSE;
2: BOOL2 := FALSE;
   BOOL3 := TRUE;
10..20: BOOL1 := TRUE;
        BOOL3:= TRUE;
ELSE
      BOOL1 := NOT BOOL1;
      BOOL2 := BOOL1 OR BOOL2;
END_CASE;
```

FOR loop

With the FOR loop one can program repeated processes.

Syntax:

```
INT_Var :INT;
FOR <INT_Var> := <INIT_VALUE> TO <END_VALUE> {BY <Step size> } DO
<Instructions>
END_FOR;
```

The part in braces {} is optional.

The <Instructions> are executed as long as the counter <INT_Var> is not greater than the <END_VALUE>. This is checked before executing the <Instructions> so that the <instructions> are never executed if <INIT_VALUE> is greater than <END_VALUE>.

When <Instructions> are executed, <INT_Var> is always increased by <Step size>. The step size can have any integer value. If it is missing, then it is set to 1. The loop must also end since <INT_Var> only becomes greater.

Example:

```
FOR Counter:=1 TO 5 BY 1 DO
Var1:=Var1*2;
```

```
END_FOR;
Erg:=Var1;
```

Let us assume that the default setting for Var1 is the value 1. Then it will have the value 32 after the FOR loop.

Note: <END_VALUE> must not be equal to the limit value of the counter <INT_VAR>. For example: If the variable Counter is of type SINT and if <END_VALUE> is 127, you will get an endless loop.

WHILE loop

The WHILE loop can be used like the FOR loop with the difference that the break-off condition can be any Boolean expression. This means you indicate a condition which, when it is fulfilled, the loop will be executed.

Syntax:

```
WHILE <Boolean expression>
<Instructions>
END_WHILE;
```

The <Instructions> are repeatedly executed as long as the <Boolean_expression> returns TRUE. If the <Boolean_expression> is already FALSE at the first evaluation, then the <Instructions> are never executed. If <Boolean_expression> never assumes the value FALSE, then the <Instructions> are repeated endlessly which causes a relative time delay.

Note: The programmer must make sure that no endless loop is caused. He does this by changing the condition in the instruction part of the loop, for example, by counting up or down one counter.

Example:

```
WHILE counter<>0 DO
  Var1 := Var1*2;
  Counter := Counter-1;
END_WHILE
```

The WHILE and REPEAT loops are, in a certain sense, more powerful than the FOR loop since one doesn't need to know the number of cycles before executing the loop. In some cases one will, therefore, only be able to work with these two loop types. If, however, the number of the loop cycles is clear, then a FOR loop is preferable since it allows no endless loops.

REPEAT loop

The REPEAT loop is different from the WHILE loop because the break-off condition is checked only after the loop has been executed. This means that the loop will run through at least once, regardless of the wording of the break-off condition.

Syntax:

```
REPEAT
<Instructions>
UNTIL <Boolean expression>
END_REPEAT;
```

The <Instructions> are carried out until the <Boolean expression> returns TRUE.

If <Boolean expression> is produced already at the first TRUE evaluation, then <Instructions> are executed only once. If <Boolean_expression> never assumes the value TRUE, then the <Instructions> are repeated endlessly which causes a relative time delay.

Note: The programmer must make sure that no endless loop is caused. He does this by changing the condition in the instruction part of the loop, for example by counting up or down one counter.

Example:

```
REPEAT
  Var1 := Var1*2;
  Counter := Counter-1;
```

```
UNTIL
  Counter=0
END_REPEAT;
```

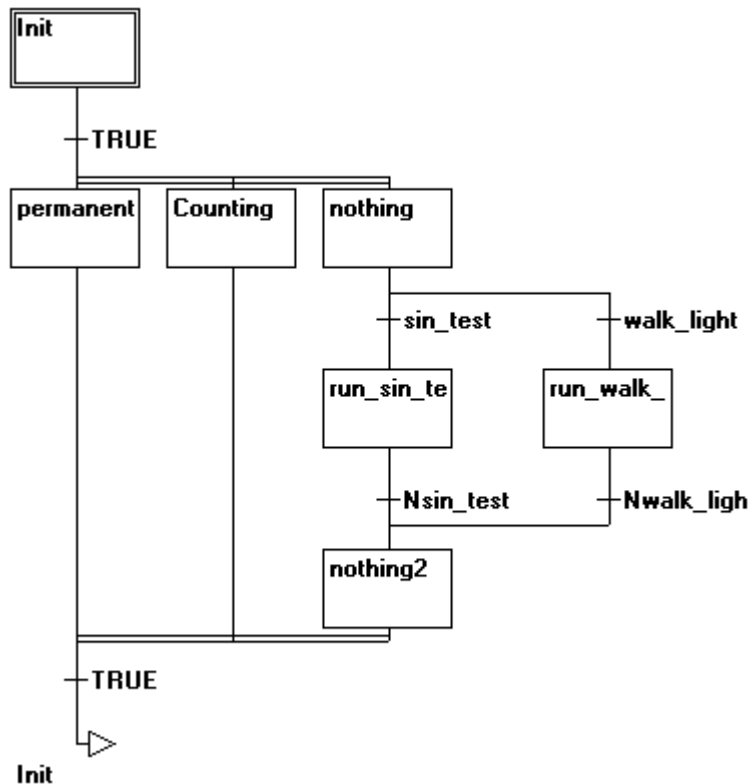
EXIT instruction

If the EXIT instruction appears in a FOR, WHILE, or REPEAT loop, then the innermost loop is ended, regardless of the break-off condition.

2.2.3 Sequential Function Chart (SFC)...

The Sequential Function Chart (SFC) is a graphically oriented language which makes it possible to describe the chronological order of different actions within a program. For this the actions are assigned to step elements and the sequence of processing is controlled by transition elements.

Example for a network in the Sequential Function Chart



For further information on the SFC Editor see Chapter 5.4.4.

Step

A POU written in a Sequential Function Chart consists of a series of steps which are connected with each other through directed connections (transitions).

There are two types of steps.

The simplified type consists of an action and a flag which shows if the step is active. If the action of a step is implemented, then a small triangle appears in upper right corner of the step.

An IEC step consists of a flag and one or more assigned actions or boolean variables. The associated actions appear to the right of the step.

Action

An action can contain a series of instructions in IL or in ST, a lot of networks in FBD or in LD, or again in Sequential Function Chart (SFC).

With the simplified steps an action is always connected to a step. In order to edit an action, click twice with the mouse on the step to which the action belongs. Or select the step and select the menu command 'Extras' 'Zoom Action/Transition'. In addition, one input or output action per step is possible.

Actions of IEC steps hang in the Object Organizer directly under their SFC-POU and are loaded with a double-click or by pressing <Enter> in their editor. New actions can be created with 'Project' 'Add Action'. You can assign max. nine actions to one IEC step.

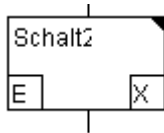
Entry or exit action

Additional to a step action you can add an entry action and an exit action to a step. An entry action is executed only once, right after the step has become active. An exit action is executed only once before the step is deactivated.

A step with entry action is indicated by an "E" in the lower left corner, the exit action by an "X" in the lower right corner.

The entry and exit action can be implemented in any language. In order to edit an entry or exit action, double-click in the corresponding corner in the step with the mouse.

Example of a step with entry and exit action:



Transition / Transition condition

Between the steps there are so-called transitions.

A transition condition must have the value TRUE or FALSE. Thus it can consist of either a boolean variable, a boolean address or a boolean constant. It can also contain a series of instructions having a boolean result, either in ST syntax (e.g. $i \leq 100$) AND b) or in any language desired (see 'Extras' 'Zoom Action/Transition'). But a transition may not contain programs, function blocks or assignments!

In the SFC-Editor a transition condition can be written directly at the transition symbol or an own editor window can be opened for entering the condition (see chapter 5.4.4, "Extras' 'Zoom Action/Transition'). Regard that the instructions entered in the editor window will take precedence!

Note: Besides transitions, inching mode can also be used to skip to the next step; see SFCtip and SFCtipmode.

Active step

After calling the SFC POU, the action (surrounded by a double border) belonging to the initial steps executed first. A step, whose action is being executed, is called active. In Online mode active steps are shown in blue.

In a control cycle all actions are executed which belong to active steps. Thereafter the respective following steps of the active steps become active if the transition conditions of the following steps are TRUE. The currently active steps will be executed in the next cycle.

Note: If the active step contains an output action, this will only be executed during the next cycle, provided that the transition following is TRUE.

IEC step

Along with the simplified steps the standard IEC steps in SFC are available.

In order to be able to use IEC steps, you must link the special SFC library **lecsfc.lib** into your project.

Not more than nine actions can be assigned to an IEC step. IEC actions are not fixed as input, step or output actions to a certain step as in the simplified steps, but are stored separately from the steps and can be reused many times within a POU. For this they must be associated to the single steps with the command 'Extras' 'Associate action'.

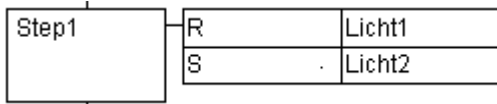
Along with actions, Boolean variables can be assigned to steps.

The activation and deactivation of actions and boolean variables can be controlled using so-called qualifiers. Time delays are possible. Since an action can still be active, if the next step has been processed, for example through the qualifier S (Set), one can achieve concurrent processes.

An associated boolean variable is set or reset with each call of the SFC block. That means, that at each call the variable gets re-assigned value TRUE resp. FALSE.

The actions associated with an IEC step are shown at the right of the step in a two-part box. The left field contains the qualifier, possibly with time constant, and the right field contains the action name respectively boolean variable name.

An example for an IEC step with two actions:



In order to make it easier to follow the processes, all active actions in online mode are shown in blue like the active steps. After each cycle a check is made to see which actions are active.

Pay attention here also to the restrictions on the use of time-qualifiers in actions that are repeatedly re-used within the same cycle (see 'Qualifier') !

Note: If an action has been inactivated, it will be executed **once more**. That means, that each action is executed at least twice (also an action with qualifier P).

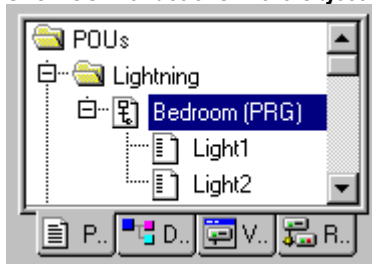
In case of a call first the deactivated actions, then the active actions are executed, in alphabetical order each time.

Whether a newly inserted step is an IEC step depends upon whether the menu command '**Extras**' '**Use IEC-Steps**' has been chosen.

In the Object Organizer the actions hang directly underneath their respective SFC POUs. New actions can be created with 'Project' 'Add Action'.

In order to use IEC steps you must include in your project the special SFC library lecsfc.lib .

SFC POU with actions in the Object Organizer



Qualifier

In order to associate the actions with IEC steps the following qualifiers are available:

- | | | |
|----------|------------------|--|
| N | Non-stored | The action is active as long as the step |
| R | overriding Reset | The action is deactivated |
| S | Set (Stored) | The action is activated and remains active until a Reset |
| L | time Limited | The action is activated for a certain time, maximum as long as the step is active |
| D | time Delayed | The action becomes active after a certain time if the step is still active and then it remains active as long as the step is active. |

P	Pulse	The action is executed just one time if the step is active
SD	Stored and Delayed	The action is activated after a certain time and remains active until a Reset
DS	Delayed and Stored	The action is activated after a certain time as long as the step is still active and remains active up to a Reset
SL	Stored and limited	The action is activated for a certain time

The qualifiers L, D, SD, DS and SL need a time value in the TIME constant format. This value can be entered directly (e.g. "T#5s") or by a variable of datatype TIME (e.g. "t_var").

Note:	When an action has been deactivated it will be executed once more . This means that each action at least is executed twice (also an action with qualifier P).
--------------	--

Implicit variables in SFC

In SFC implicitly declared variables ("flags") can be used to scan the status of steps and actions as well as the time of steps. These flags each are set at the beginning of a cycle. For IEC steps and IEC actions they are provided by the library iecsfc.lib (structures SFCStepType and SFCActionType), which is included automatically in a project, for the simplified steps they are implemented in CoDeSys.

Scan of the step or action status via boolean variables:

For IEC steps: <stepname>.x resp. <stepname>._x: <StepName>.x shows the current activation status. <StepName>._x shows the activation status for the next cycle. If <StepName>.x=TRUE, the step will be executed in the current cycle. If <StepName>._x=TRUE and <StepName>.x=FALSE, the step will be executed in the following cycle, i.e. <StepName>._x gets copied to <StepName>.x at the beginning of a cycle.

For simplified steps: <stepname> resp. _<stepname>: If <StepName>=TRUE, the step will be executed in the current cycle. If _<StepName>=TRUE, the step will be executed in the following cycle, i.e. <StepName> gets copied to _<StepName> at the beginning of a cycle.

For IEC-actions: <actionname>.x gets TRUE as soon as the action gets active (<actionname>._x is only for internal purposes, not for a status scan).

Time of a step via TIME variables:

The following implicit variables give the current time span which has passed since the step had got active; this is only for steps which have a minimum time configured in the step attributes.

For IEC steps: <stepname>.t (<stepname>._t not usable for external purposes)

For simplified steps: _time<stepname>. BUT: If this implicit variable should be used for scan purposes, it also must be declared explicitly as a TIME variable; e.g. "_timeStep1 : TIME;"

For IEC actions: the implicit time variables are not used.

These status flags can be used in each action and transition of the SFC module. But they can also be accessed from other programs: Example: boolvar1:=sfc.step1.x; step1.x in this example is an implicit boolean variable showing the status of IEC step "step1" in POU "sfc1".

SFC Flags

For controlling the operation of SFC POU's flags can be used, which are created implicitly during running the project. To read this flags you have to define appropriate global or local variables as inputs or outputs. Example: If in a SFC POU a step is active for a longer time than defined in the step attributes, then a flag will be set, which is accessible by using a variable "SFCErrror" (SFCErrror gets TRUE in this case).

The following flag variables can be defined:

SFCEnableLimit: This variable is of the type BOOL. When it has the value TRUE, the timeouts of the steps will be registered in SFCError. Other timeouts will be ignored.

SFCInit: When this boolean variable has the value TRUE the sequential function chart is set back to the Init step. The other SFC flags are reset too (initialization). The Init step remains active, but is not executed, for as long as the variable has the value TRUE. It is only when SFCInit is again set to FALSE that the block can be processed normally.

SFCReset: This variable, of type BOOL, behaves similarly to SFCInit. Unlike the latter, however, further processing takes place after the initialization of the Init step. Thus for example the SFCReset flag could be re-set to FALSE in the Init step.

Please regard: As from compiler version 2.3.7.0 SFCReset also can be used to reset boolean actions associated to IEC steps, which was not possible before

SFCQuitError: Provided that the Execution of the SFC diagram is stopped for as long as this boolean variable has the value TRUE whereby a possible timeout in the variable SFCError is reset. All previous times in the active steps are reset when the variable again assumes the value FALSE. It is a pre-condition that the flag SFCError has been defined also, which registers any timeout in the SFC.

SFCPause: Execution of the SFC diagram is stopped for as long as this boolean variable has the value TRUE.

SFCError: This Boolean variable is TRUE when a timeout has occurred in a SFC diagram. If another timeout occurs in a program after the first one, it will not be registered unless the variable SFCError is reset first. It is a pre-condition that SFCError is defined, if you want to use the other time-controlling flags (SFCErrorStep, SFCErrorPOU, SFCQuitError, SFCErrorAnalyzation).

SFCTrans: This boolean variable takes on the value TRUE when a transition is actuated.

SFCErrorStep: This variable is of the type STRING. If SFCError registers a timeout, in this variable is stored the name of the step which has caused the timeout. It is a pre-condition that the flag SFCError has been defined also, which registers any timeout in the SFC.

SFCErrorPOU: This variable of the type STRING contains the name of the block in which a timeout has occurred. It is a pre-condition that the flag SFCError has been defined also, which registers any timeout in the SFC.

SFCCurrentStep: This variable is of the type STRING. The name of the step is stored in this variable which is active, independently of the time monitoring. In the case of simultaneous sequences the step is stored in the branch on the outer right. No further timeout will be registered if a timeout occurs and the variable SFCError is not reset again.

SFCErrorAnalyzationTable: This variable of type ARRAY [0..n] OF ExpressionResult provides the result of an analyzation of a transition expression. For each component of the expression, which is contributing to a FALSE of the transition and thereby to a timeout of the preceding step, the following information is written to the structure ExpressionResult: name, address, comment, current value.

This is possible for maximum 16 components (variables), thus the array range is max. 0..15).

The structure ExpressionResult as well as the implicitly used analyzation modules are provided with the library AnalyzationNew.lib. The analyzation modules also can be used explicitly in other POU's, which are not programmed in SFC.

It is a pre-condition for the analyzation of a transition expression, that a timeout is registered in the preceding step. So a time monitoring must be implemented there and also the variable SFCError (see above) must be defined in the declaration window.

SFCTip, SFCTipMode: This variables of type BOOL allow inching mode of the SFC. When this is switched on by SFCTipMode=TRUE, it is only possible to skip to the next step if SFCTip is set to TRUE. As long as SFCTipMode is set to FALSE, it is possible to skip even over transitions.

Note: Regard also the implicit variables usable for scanning the status and time of steps resp. actions.

Alternative branch

Two or more branches in SFC can be defined as alternative branches. Each alternative branch must begin and end with a transition. Alternative branches can contain parallel branches and other alternative branches. An alternative branch begins at a horizontal line (alternative beginning) and ends at a horizontal line (alternative end) or with a jump.

If the step which precedes the alternative beginning line is active, then the first transition of each alternative branch is evaluated from left to right. The first transition from the left whose transition condition has the value TRUE is opened and the following steps are activated (see active step).

Parallel branch

Two or more branches in SFC can be defined as parallel branches. Each parallel branch must begin and end with a step. Parallel branches can contain alternative branches or other parallel branches. A parallel branch begins with a double line (parallel beginning) and ends with a double line (parallel end) or with a jump. It can be provided with a jump label.

If the parallel beginning line of the previous step is active and the transition condition after this step has the value TRUE, then the first steps of all parallel branches become active (see active step). These branches are now processed parallel to one another. The step after the parallel end line becomes active when all previous steps are active and the transition condition before this step produces the value TRUE.

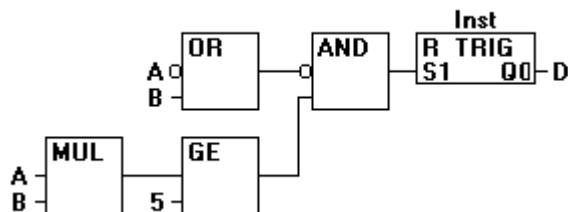
Jump

A jump is a connection to the step whose name is indicated under the jump symbol. Jumps are required because it is not allowed to create connections which lead upward or cross each other.

2.2.4 Function Block Diagram (FBD)...

The Function Block Diagram is a graphically oriented programming language. It works with a list of networks whereby each network contains a structure which represents either a logical or arithmetic expression, the call of a function block, a jump, or a return instruction.

Example of a network in the Function Block Diagram



For further information on the FBD editor see Chapter 5.4.2.

2.2.5 The Continuous Function Chart Editor (CFC)...

The continuous function chart editor does not operate like the function block diagram FBD with networks, but rather with freely placeable elements. This allows feedback, for example.

For further information on the CFC editor see Chapter 5.4.5

Example of a network in the continuous function chart editor



2.2.6 Ladder Diagram (LD)...

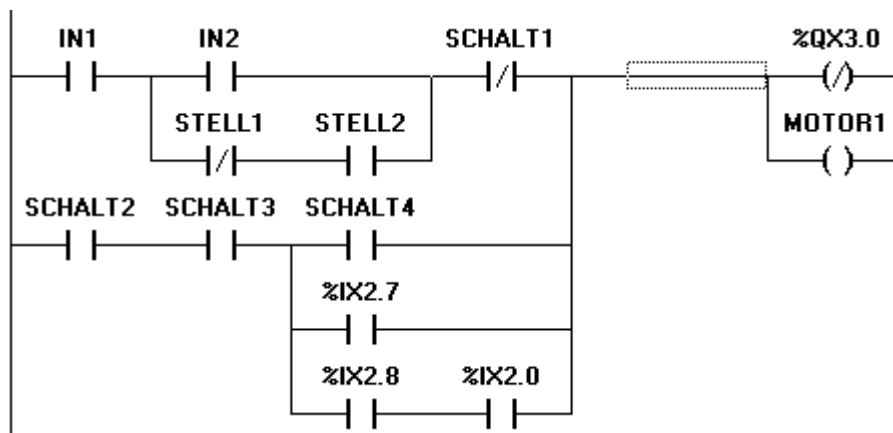
The Ladder Diagram is also a graphics oriented programming language which approaches the structure of an electric circuit.

On the one hand, the Ladder Diagram is suitable for constructing logical switches, on the other hand one can also create networks as in FBD. Therefore the LD is very useful for controlling the call of other POUs.

The Ladder Diagram consists of a series of networks. A network is limited on the left and right sides by a left and right vertical current line. In the middle is a circuit diagram made up of contacts, coils, and connecting lines.

Each network consists on the left side of a series of contacts which pass on from left to right the condition "ON" or "OFF" which correspond to the Boolean values TRUE and FALSE. To each contact belongs a Boolean variable. If this variable is TRUE, then the condition is passed from left to right along the connecting line. Otherwise the right connection receives the value OFF.

Example of a network in a Ladder Diagram made up of contacts and coils



For further information on the LD editor see Chapter 5.4.3.

Contact

Each network in LD consists on the left side of a network of *contacts* (contacts are represented by two parallel lines: | |) which from left to right show the condition "On" or "Off".

These conditions correspond to the Boolean values TRUE and FALSE. A Boolean variable belongs to each contact. If this variable is TRUE, then the condition is passed on by the connecting line from left to right, otherwise the right connection receives the value "Out".

Contacts can be connected in parallel, then one of the parallel branches must transmit the value "On" so that the parallel branch transmits the value "On"; or the contacts are connected in series, then contacts must transmit the condition "On" so that the last contact transmits the "On" condition. This therefore corresponds to an electric parallel or series circuit.

A contact can also be negated, recognizable by the slash in the contact symbol: |/. Then the value of the line is transmitted if the variable is FALSE.

Coil

On the right side of a network in LD there can be any number of so-called coils which are represented by parentheses:(). They can only be in parallel. A coil transmits the value of the connections from left to right and copies it in an appropriate Boolean variable. At the entry line the value ON (corresponds to the Boolean variable TRUE) or the value OFF (corresponding to FALSE) can be present.

Contacts and coils can also be negated (in the example the contact SWITCH1 and the coil %QX3.0 is negated). If a coil is negated (recognizable by the slash in the coil symbol: (/)), then it copies the negated value in the appropriate Boolean variable. If a contact is negated, then it connects through only if the appropriate Boolean value is FALSE.

Function blocks in the Ladder Diagram

Along with contacts and coils you can also enter function blocks and programs in the network. They must have an input and an output with Boolean values and can be used at the same places as contacts, that is on the left side of the LD network.

Set/Reset coils

Coils can also be defined as set or reset coils. One can recognize a set coil by the "S" in the coil symbol: **(S)**. It never writes over the value TRUE in the appropriate Boolean variable. That is, if the variable was once set at TRUE, then it remains so.

One can recognize a reset coil by the "R" in the coil symbol: **(R)**. It never writes over the value FALSE in the appropriate Boolean variable: If the variable has been once set on FALSE, then it remains so.

LD as FBD

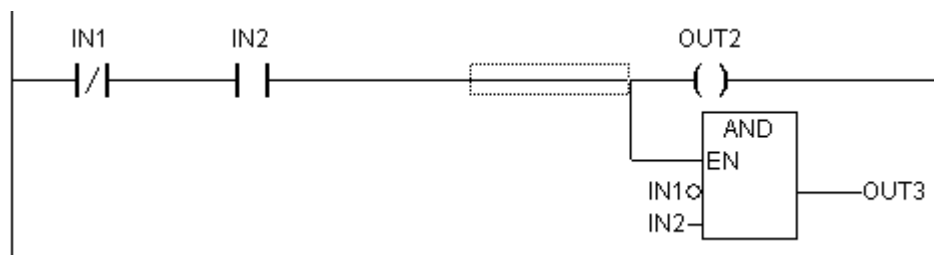
When working with LD it is very possible that you will want to use the result of the contact switch for controlling other POU's. On the one hand you can use the coils to put the result in a global variable which can then be used in another place. You can, however, also insert the possible call directly into your LD network. For this you introduce a POU with EN input.

Such POU's are completely normal operands, functions, programs, or function blocks which have an additional input which is labeled with EN. The EN input is always of the BOOL type and its meaning is: The POU with EN input is evaluated when EN has the value TRUE.

An EN POU is wired parallel to the coils, whereby the EN input is connected to the connecting line between the contacts and the coils. If the ON information is transmitted through this line, this POU will be evaluated completely normally.

Starting from such an EN POU, you can create networks similar to FBD.

Example of a LD network with an EN POU



2.3 Debugging, Online Functions...

Sampling Trace

The Sampling Trace allows you to trace the value sequence of variables, depending upon the so-called trigger event. This is the rising edge or falling edge of a previously defined Boolean variable (trigger variable). CoDeSys permits the tracing of up to 20 variables. 500 values can be traced for each variable.

Debugging

The debugging functions of CoDeSys make it easier for you to find errors.

In order to debug, run the command **'Project' 'Options'** and in the dialog box that pops up under Build options select activate option **Debugging**.

Breakpoint

A breakpoint is a place in the program at which the processing is stopped. Thus it is possible to look at the values of variables at specific places within the program.

Breakpoints can be set in all editors. In the text editors breakpoints are set at line numbers, in FBD and LD at network numbers, in CFC at POUs and in SFC at steps. No breakpoints can be set in function block instances.

Attention: Runtime system CoDeSys SP 32 Bit Full will deactivate the watchdog function of the concerned task as soon as the execution of the program currently is stopped at a breakpoint.

Single step

Single step means:

In IL: Execute the program until the next CAL, LD or JMP command.

In ST: Execute the next instruction.

□□□ In FBD, LD: Execute the next network.

□□□ In SFC: Continue the action until the next step.

By proceeding step by step you can check the logical correctness of your program.

Single Cycle

If Single cycle has been chosen, then the execution is stopped after each cycle.

Change values online

During operations variables can be set once at a certain value (write value) or also described again with a certain value after each cycle (force value). In online mode one also can change the variable value by double click on the value. By that boolean variables change from TRUE to FALSE or the other way round, for each other types of variables one gets the dialog **Write Variable xy**, where the actual value of the variable can be edited.

Monitoring

In Online mode, all displayable variables are read from the controller and displayed in real time. You will find this display in the declarations and program editor; you can also read out current values of variables in the watch and receipt manager and in a visualization. If variables from instances of function blocks are to be monitored, the corresponding instance must first be opened.

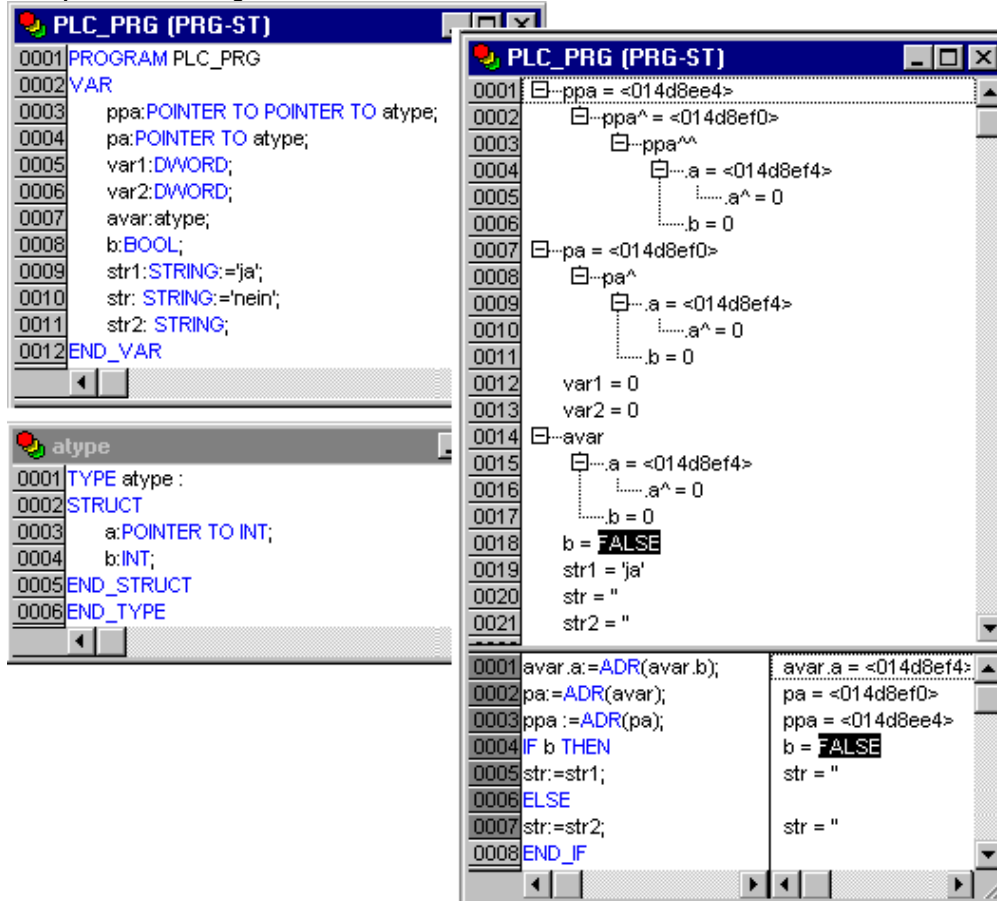
In monitoring VAR_IN_OUT variables, the de-referenced value is output.

In monitoring pointers, both the pointer and the de-referenced value are output in the declaration portion. In the program portion, only the pointer is output:

```
+ --pointervar = '<'pointervalue>'
```

POINTERS in the de-referenced value are also displayed accordingly. With a simple click on the cross or a double-click on the line, the display is either expanded or truncated.

Example for Monitoring of Pointers



In the implementations, the value of the pointer is displayed. For de-referencing, however, the de-referenced value is displayed.

Monitoring of ARRAY components: In addition to array components indexed by a constant, components are also displayed which are indexed by a variable:

anarray[1] = 5
 anarray[i] = 1

If the index consists of an expression (e.g. [i+] or [i+1]), the component can not be displayed.

Please regard: If the maximum number of variables which can be monitored, has been reached, for each further variable instead of the current value the string "Too many monitoring variables" will be displayed.

Simulation

During the simulation the created PLC program is not processed in the PLC, but rather in the calculator on which **CoDeSys** is running. All online functions are available. That allows you to test the logical correctness of your program without PLC hardware.

Please regard: POU's of external libraries do not run in simulation mode.

Log

The log chronologically records user actions, internal processes, state changes and exceptions during Online mode processing. It is used for monitoring and for error tracing (see Online Functions).

2.4 The Standard...

The standard IEC 61131-3 is an international standard for programming languages of Programmable Logic Controllers.

The programming languages offered in **CoDeSys** conform to the requirements of the standard.

According to this standard, a program consists of the following elements:

Structures (see Data Types)

POUs

Global Variables

The general language elements are described in the sections Identifier, Addresses, Types, Comments, and Constants.

The processing of a **CoDeSys** program starts with the special POU PLC_PRG. The POU PLC_PRG can call other POUs.

3 We Write a Little Program

3.1 Controlling a Traffic Signal Unit...

Let us now start to write a small example program. It is for a simple traffic signal unit which is supposed to control two traffic signals at an intersection. The red/green phases of both traffic signals alternate and, in order to avoid accidents, we will insert yellow or yellow/red transitional phases. The latter will be longer than the former.

In this example you will see how time dependent programs can be shown with the language resources of the IEC1131-3 standard, how one can edit the different languages of the standard with the help of **CoDeSys**, and how one can easily connect them while becoming familiar with the simulation of **CoDeSys**.

Create POU

Starting always is easy: Start **CoDeSys** and choose 'File' 'New'.

In the dialog box which appears, the first POU has already been given the default name PLC_PRG. Keep this name, and the type of POU should definitely be a program. Each project needs a program with this name. In this case we choose as the language of this POU the Continuous Function Chart Editor (CFC)

Now create three more objects with the command 'Project' 'Object Add' with the menu bar or with the context menu (press right mouse button in the Object Organizer). A program in the language Sequential Function Chart (SFC) named SEQUENCE, a function block in the language Function Block Diagram (FBD) named TRAFFICSIGNAL, along with a POU WAIT, also of the type function block, which we want to program as an Instruction List (IL).

What does TRAFFICSIGNAL do?

In the POU TRAFFICSIGNAL we will assign the individual trafficsignal phases to the lights, i.e. we will make sure that the red light is lit red in the red phase and in the yellow/red phase, the yellow light in the yellow and yellow/red phases, etc.

What does WAIT do?

In WAIT we will program a simple timer which as input will receive the length of the phase in milliseconds, and as output will produce TRUE as soon as the time period is finished.

What does SEQUENCE do?

In SEQUENCE all is combined so that the right light lights up at the right time for the desired time period.

What does PLC_PRG do?

In PLC_PRG the input start signal is connected to the traffic lights' sequence and the "color instructions" for each lamp are provided as outputs.

TRAFFICSIGNAL simulation

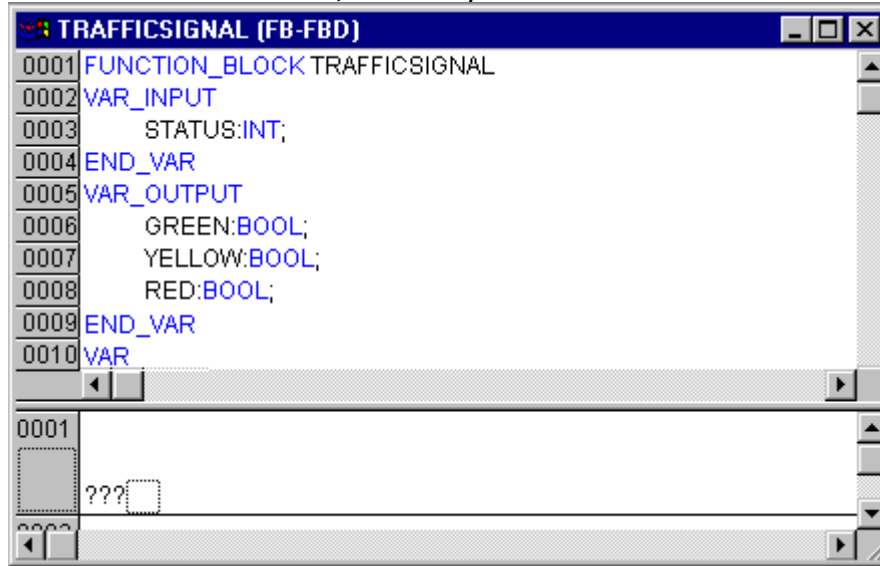
Now test your program in simulation mode. Compile ('Project' 'Build') and load ('Online' 'Login') it. Start the program by 'Online' 'Start', then set variable ON to TRUE, e.g. by a double-click on the entry "ON" in the input box of the CFC editor. This will mark the variable as prepared to be set to <TRUE>. Then press <Ctrl><F7> or command 'Online' 'Write values', to set the value. Now variable START in ABLAUF (which we had set to TRUE manually in the first extension level of the program) gets this value by variable ON, which is used in PLC_PRG. This will make run the traffic light cycles. PLC_PRG has changed to a monitoring window. Click twice on the plus sign in the declaration editor, the variable display drops down, and you can see the values of the individual variables.

"TRAFFICSIGNAL" declaration

Let us now turn to the POU TRAFFICSIGNAL. In the declaration editor you declare as input variable (between the keywords VAR_INPUT and END_VAR) a variable named STATUS of the type INT. STATUS will have four possible conditions, that is one for the TRAFFICSIGNAL phases green, yellow, yellow/red and red.

Correspondingly our TRAFFICSIGNAL has three outputs, that is RED, YELLOW and GREEN. You should declare these three variables. Then the declaration part of our function block TRAFFICSIGNAL will look like this:

Function block TRAFFICSIGNAL, declaration part



"TRAFFICSIGNAL" body

Now we determine the values of the output variables depending on the input STATUS of the POU. To do this go into the body of the POU. Click on the field to the left beside the first network (the gray field with the number 0001). You have now selected the first network. Choose the menu item 'Insert' 'Box'.

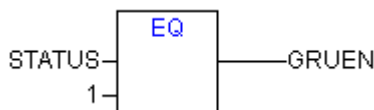
In the first network a box is inserted with the operator AND and two inputs:



Click on the text AND, so that it appears selected and change the text into EQ. Select then for each of the two inputs the three question marks and overwrite them with "STATUS" respectively "1".



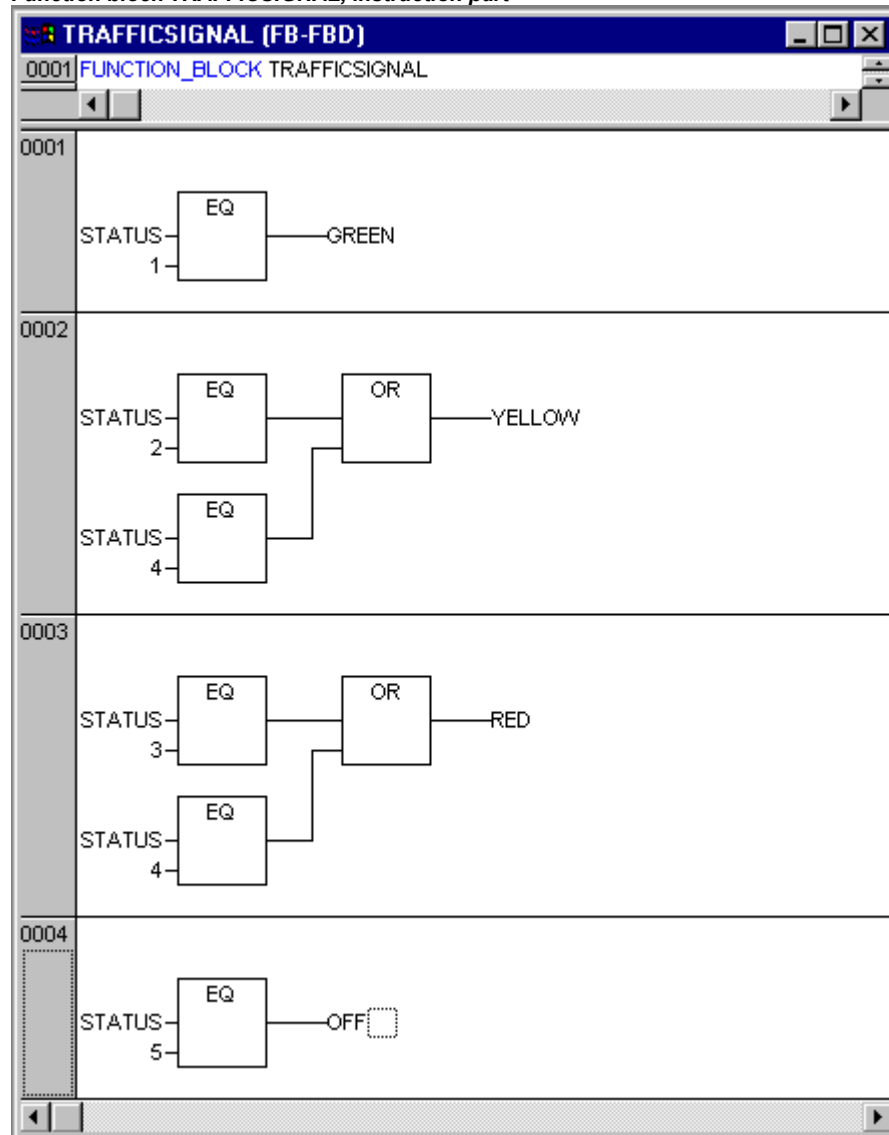
Click now on a place behind the EQ Box. Now the output of the EQ operation is selected. Choose 'Insert' 'Assign'. Change the three question marks ??? to GREEN. You now have created a network with the following structure:



STATUS is compared with 1, the result is assigned to GREEN. This network thus switches to GREEN if the preset state value is 1.

For the other TRAFFICSIGNAL colors we need two more networks. To create the first one execute command **'Insert' 'Network (after)'** and insert an EQ-Box like described above. Then select the output pin of this box and use again command **'Insert' 'Box'**. In the new box replace "AND" by "OR". Now select the first output pin of the OR-box and use command **'Insert' 'Assign'** to assign it to "GELB". Select the second input of the OR-box by a mouse-click on the horizontal line next to the three question marks, so that it appears marked by a dotted rectangle. Now use **'Insert' 'Box'** to add a further EQ-box like described above. Finally the network should look like shown in the following:

Function block TRAFFICSIGNAL, instruction part



In order to insert an operator in front of another operator, you must select the place where the input to which you want to attach the operator feeds into the box.

Then use the command **'Insert' 'Box'**. Otherwise you can set up these networks in the same way as the first network.

Now our first POU has been finished. TRAFFICSIGNAL, according to the input of the value STATUS, controls whichever light color we wish.

Connecting the standard.lib

For the timer in the POU WAIT we need a POU from the standard library. Therefore, open the library manager with **'Window' 'Library Manager'**. Choose **'Insert' 'Additional library'**. The dialog box appears for opening files. From the list of the libraries choose standard.lib.

"WAIT" declaration

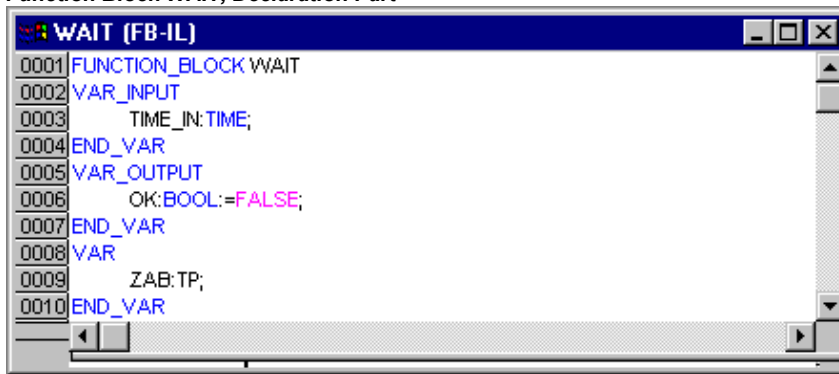
Now let us turn to the POU WAIT. This POU is supposed to become a timer with which we can determine the length of the time period of each TRAFFICSIGNAL phase. Our POU receives as input variable a variable TIME of the type TIME, and as output it produces a Boolean value which we want to call OK and which should be TRUE when the desired time period is finished. We set this value with FALSE by inserting at the end of the declaration (before the semicolon, however) " := FALSE ".

For our purposes we need the POU TP, a clock generator. This has two inputs (IN, PT) and two outputs (Q, ET). TP does the following:

As long as IN is FALSE, ET is 0 and Q is FALSE. As soon as IN provides the value TRUE, the time is calculated at the output ET in milliseconds. When ET reaches the value PT, then ET is no longer counted. Meanwhile Q produces TRUE as long as ET is smaller than PT. As soon as the value PT has been reached, then Q produces FALSE again. See the chapter on the standard library for short descriptions of all POUs.

In order to use the POU TP in the POU WAIT we must create a local instance from TP. For this we declare a local variable ZAB (for elapsed time) of the type TP (between the keywords VAR, END_VAR).

The declaration part of WAIT thus looks like this:

Function Block WAIT, Declaration Part


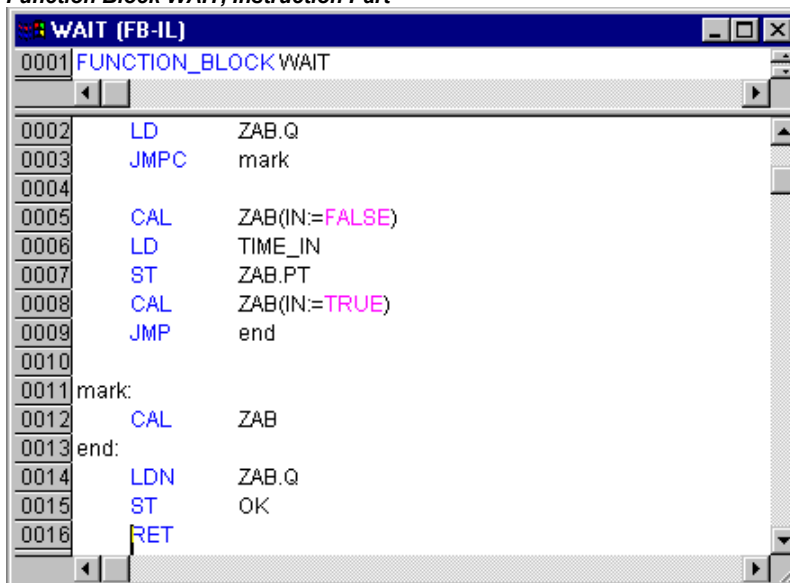
```

0001 FUNCTION_BLOCK WAIT
0002 VAR_INPUT
0003     TIME_IN:TIME;
0004 END_VAR
0005 VAR_OUTPUT
0006     OK:BOOL:=FALSE;
0007 END_VAR
0008 VAR
0009     ZAB:TP;
0010 END_VAR

```

"WAIT" body

In order to create the desired timer, the body of the POU must be programmed as follows:

Function Block WAIT, Instruction Part


```

0001 FUNCTION_BLOCK WAIT
0002 LD ZAB.Q
0003 JMPC mark
0004
0005 CAL ZAB(IN:=FALSE)
0006 LD TIME_IN
0007 ST ZAB.PT
0008 CAL ZAB(IN:=TRUE)
0009 JMP end
0010
0011 mark:
0012 CAL ZAB
0013 end:
0014 LDN ZAB.Q
0015 ST OK
0016 RET

```

At first it is checked whether Q has already been set at TRUE (as though the counting had already been executed), in this case we change nothing with the occupation of ZAB, but we call the function block ZAB without input (in order to check whether the time period is already over).

Otherwise we set the variable IN in ZAB at FALSE, and therefore at the same time ET at 0 and Q at FALSE. In this way all variables are set at the desired initial condition. Now we assign the necessary time from the variable TIME into the variable PT, and call ZAB with IN:=TRUE. In the function block ZAB the variable ET is now calculated until it reaches the value TIME, then Q is set at FALSE.

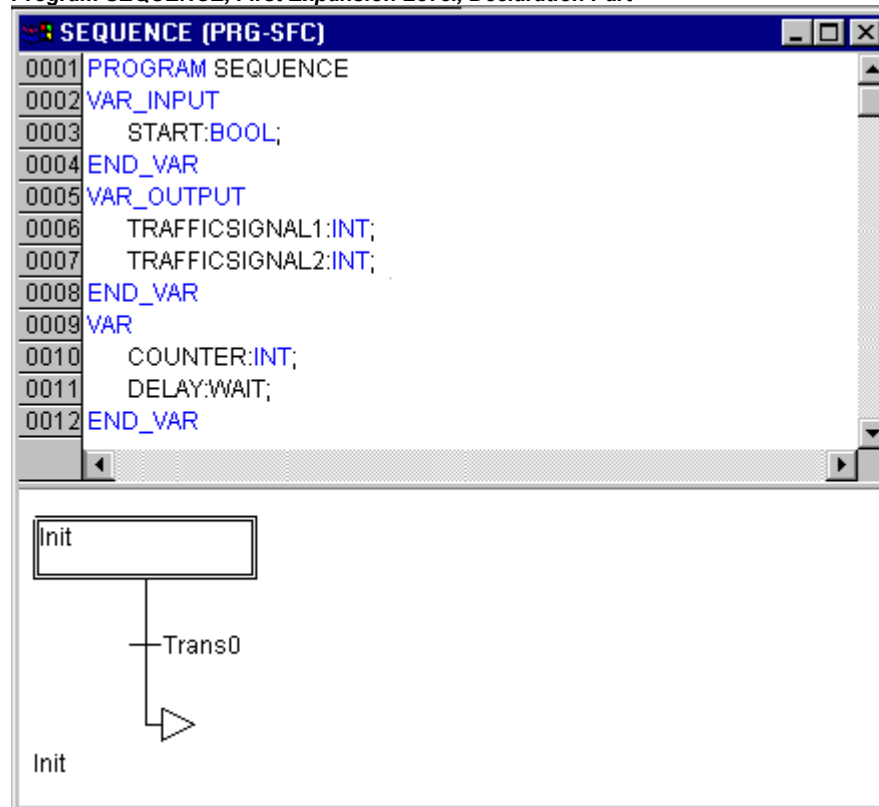
The negated value of Q is saved in OK after each execution of WAIT. As soon as Q is FALSE, then OK produces TRUE.

The timer is finished at this point. Now it is time to combine our two function blocks WAIT and SEQUENCE in the main program PLC_PRG.

"SEQUENCE" first expansion level

First we declare the variables we need. They are: an input variable START of the type BOOL, two output variables TRAFFICSIGNAL1 and TRAFFICSIGNAL2 of the type INT and one of the type WAIT (DELAY as delay). The program SEQUENCE now looks like shown here:

Program SEQUENCE, First Expansion Level, Declaration Part



Create a SFC diagram

The beginning diagram of a POU in SFC always consists of an action "Init" of a following transition "Trans0" and a jump back to Init. We have to expand that.

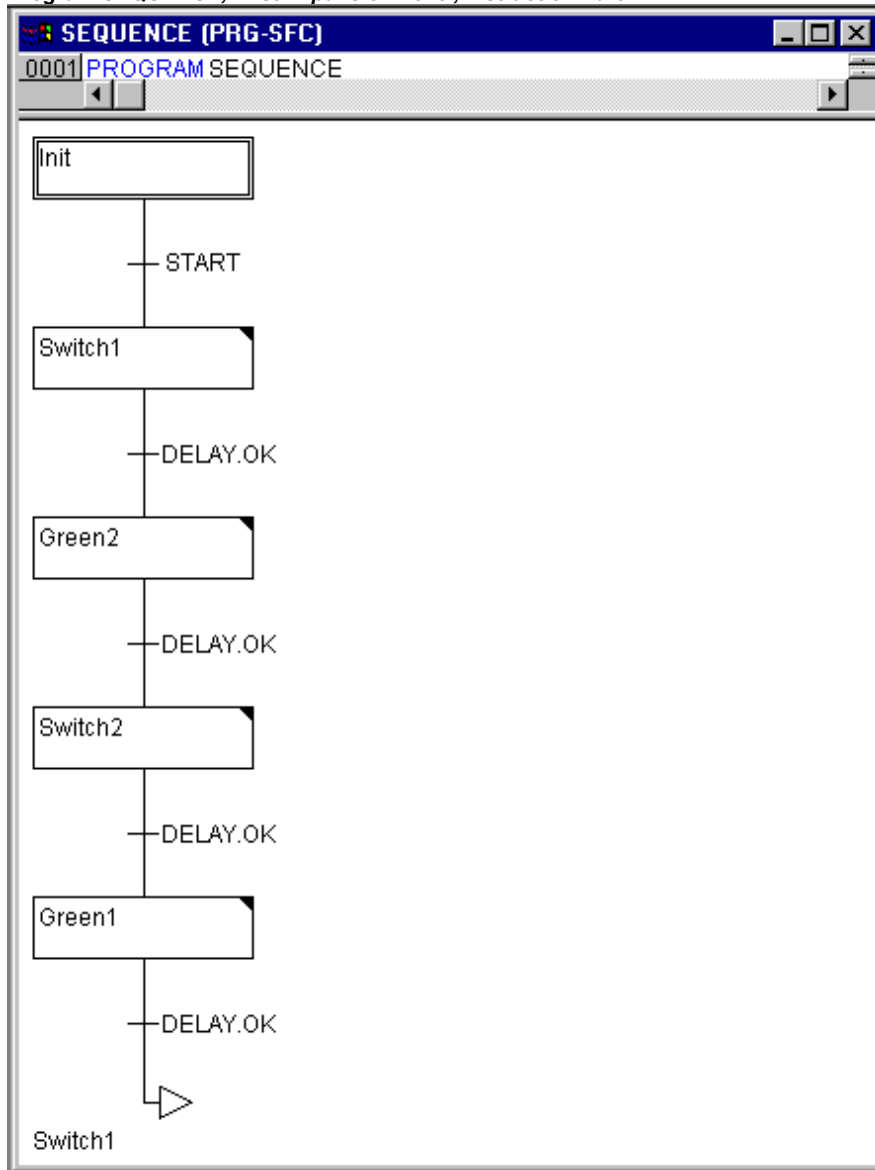
Before we program the individual action and transitions let us first determine the structure of the diagrams. We need one step for each TRAFFICSIGNAL phase. Insert it by marking Trans0 and choosing 'Insert' 'Step transition (after)'. Repeat this procedure three more times.

If you click directly on the name of a transition or a step, then this is marked and you can change it. Name the first transition after Init "START", and all other transitions "DELAY.OK".

The first transition switches through when START is TRUE and all others switch through when DELAY in OK produces TRUE, i.e. when the set time period is finished.

The steps (from top to bottom) receive the names Switch1, Green2, Switch2, Green1, whereby Init of course keeps its Name. "Switch" should include a yellow phase, at Green1 TRAFFICSIGNAL1 will be green, at Green2 TRAFFICSIGNAL2 will be green. Finally change the return address of Init after Switch1. If you have done everything right, then the diagram should look like in the following image:

Program SEQUENCE, First Expansion Level, Instruction Part



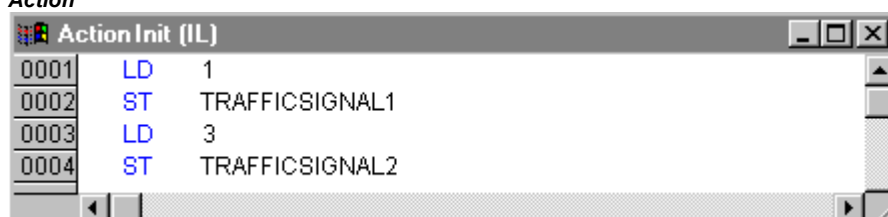
Now we have to finish the programming of the individual steps. If you double-click on the field of a step, then you get a dialog for opening a new action. In our case we will use IL (Instruction List).

Actions and transition conditions

In the action of the step **Init** the variables are initialized, the STATUS of TRAFFICSIGNAL1 should be 1 (green). The state of TRAFFICSIGNAL2 should be 3 (red). The action Init then looks like in the following image:

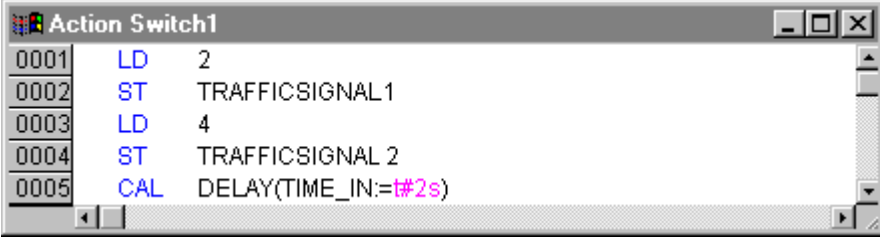
Action

Init



Switch1 changes the state of TRAFFICSIGNAL1 to 2 (yellow), and that of TRAFFICSIGNAL2 to 4 (yellow-red). In addition, a time delay of 2000 milliseconds is set. The action is now as follows:

Action Switch1



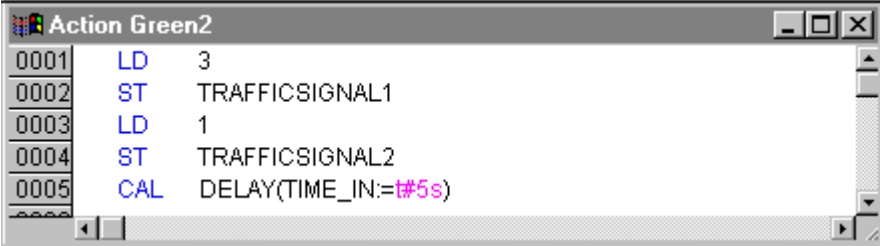
```

0001 LD 2
0002 ST TRAFFICSIGNAL1
0003 LD 4
0004 ST TRAFFICSIGNAL2
0005 CAL DELAY(TIME_IN:=#2s)

```

With **Green2** TRAFFICSIGNAL1 is red (STATUS:=3), TRAFFICSIGNAL2 green (STATUS:=1), and the delay time is 5000 milliseconds.

Action Green2



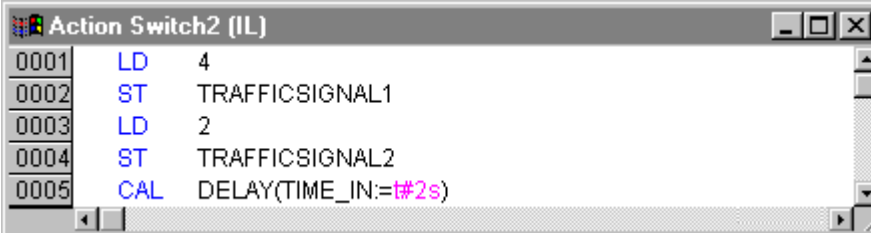
```

0001 LD 3
0002 ST TRAFFICSIGNAL1
0003 LD 1
0004 ST TRAFFICSIGNAL2
0005 CAL DELAY(TIME_IN:=#5s)

```

At **Switch2** the STATUS of TRAFFICSIGNAL1 changes to 4 (yellow-red), that of TRAFFICSIGNAL2 to 2 (yellow). A time delay of 2000 milliseconds is now set.

Action Switch2



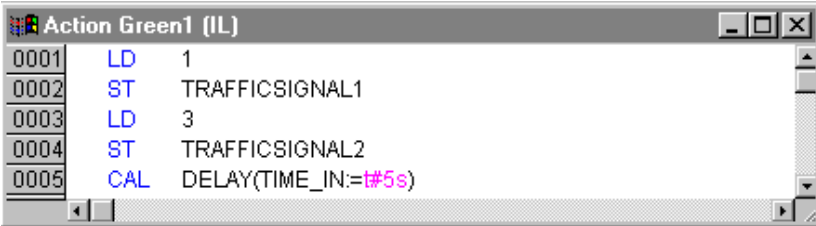
```

0001 LD 4
0002 ST TRAFFICSIGNAL1
0003 LD 2
0004 ST TRAFFICSIGNAL2
0005 CAL DELAY(TIME_IN:=#2s)

```

With **Green1** TRAFFICSIGNAL1 is green (STATUS:=1), TRAFFICSIGNAL2 is red (STATUS:=3), and the time delay is set to 5000 milliseconds.

Action Green1



```

0001 LD 1
0002 ST TRAFFICSIGNAL1
0003 LD 3
0004 ST TRAFFICSIGNAL2
0005 CAL DELAY(TIME_IN:=#5s)

```

The first expansion phase of our program is completed.

If you want to do a first test of POU ABLAUF in simulation mode, perform the following steps:

1. Open POU PLC_PRG. Each project starts running with PLC_PRG. In order to be able to provisionally start POU ABLAUF, insert a box and replace "AND" by "ABLAUF". Remain the inputs and outputs unassigned for the moment.
2. Compile the project via 'Project' 'Build'. In the message window you should get "0 Errors, 0 Warnings". Now check if option 'Online' 'Simulation' is activated and use command 'Online' 'Login' to get into simulation mode. Start program with 'Online' 'Start'. Open POU ABLAUF by a double-click on entry "ABLAUF" in the Object Organizer. The program is started now, but to get it run, variable START must be TRUE. Later this will be set by PLC_PRG but at the moment we have to set it manually within

the POU. To do that, perform a double-click on the line in the declaration part, where START is defined (START=FALSE). This will set the option "<:=TRUE>" behind the variable in turquoise color. Now select command 'Online' 'Write values' to set this value. Thereupon START will be displayed in blue color in the sequence diagram and the processing of the steps will be indicated by a blue mark of the currently active step.

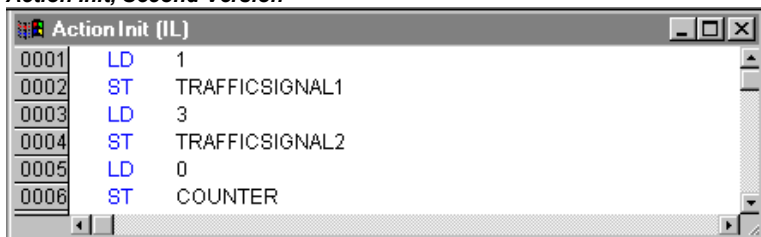
When you have finished this intermediate test use command 'Online' 'Logout' to leave the simulation mode and to continue programming.

"SEQUENCE" second expansion level

In order to ensure that our diagram has at least one alternative branch, and so that we can turn off our traffic light unit at night, we now include in our program a counter which, after a certain number of TRAFFICSIGNAL cycles, turns the unit off.

At first we need a new variable COUNTER of the type INT. Declare this as usual in the declaration part of SEQUENCE, and initialize it in Init with 0.

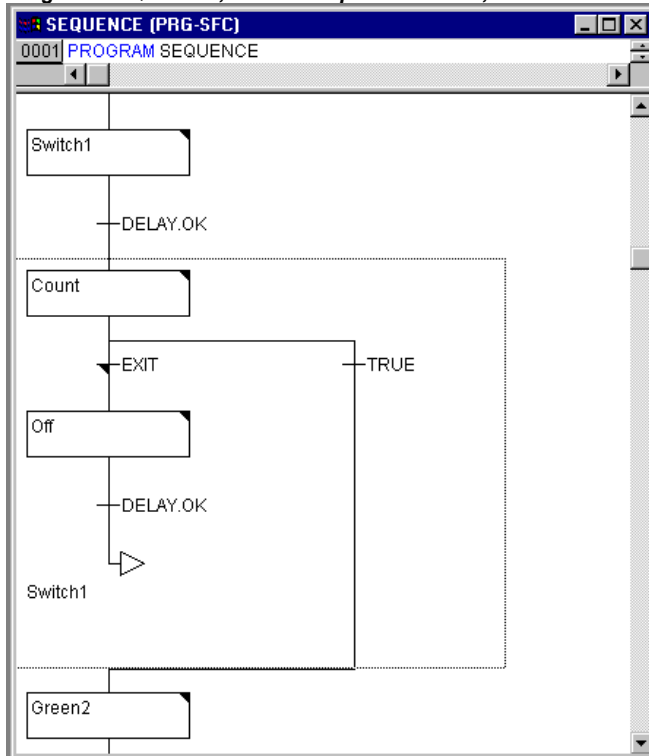
Action Init, Second Version



Now select the transition after Switch1 and insert a step and then a transition. Select the resulting transition and insert an alternative branch to its left. After the left transition insert a step and a transition. After the resulting new transition insert a jump after Switch1.

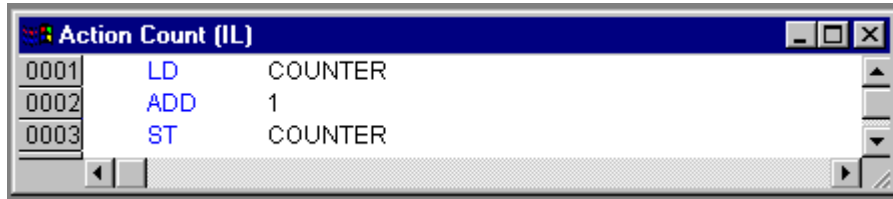
Name the new parts as follows: the upper of the two new steps should be called "Count" and the lower "Off". The transitions are called (from top to bottom and from left to right) EXIT, TRUE and DELAY.OK. The new part should look like the part marked with the black border in the following image:

Program SEQUENCE, Second Expansion Level, Instruction Part



Now two new actions and a new transition condition are to be implemented. At the step Count the variable COUNTER is increased by one:

Action Count



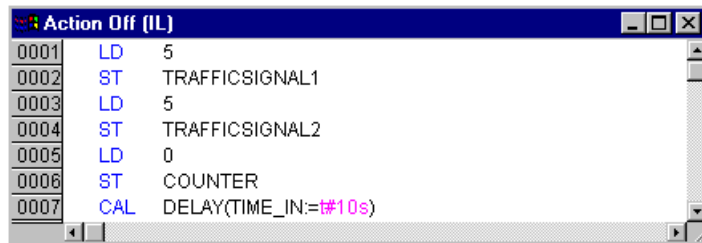
The EXIT transition checks whether the counter is greater than a certain value, for example 7:

Transition EXIT



At Off the state of both lights is set at 5(OFF), (or each other number not equal 1,2,3 or 4) the COUNTER is reset to 0, and a time delay of 10 seconds is set:

Action Off



The result

In our hypothetical situation, night falls after seven trafficsignal cycles, for ten seconds the trafficsignal turns itself off, then we have daylight again, the traffic light unit turns itself on again, and the whole process starts again from the beginning. If you like, do another test of the current version of your program in simulation mode before we go on to create the POU PLC_PRG.

PLC_PRG

We have defined and correlated the time sequencing of the phases for both sets of traffic lights in the block SEQUENCE. Since, however, we see the traffic lights system as a module of a bus system, e.g. CAN bus, we have to make input and output variables available in the block PLC_PRG. We want to start-up the traffic lights system over an ON switch and we want to send each of the six lamps (each traffic light red, green, yellow) the corresponding "signal command" for each step of the SEQUENCE. We are now declaring appropriate Boolean variables for these six outputs and one input, before we create the programme in the editor, and are allocating them, at the same time, to the corresponding IEC addresses.

The next step is declare the variables Light1 and Light2 of the type Phases in the declaration editor.

Declaration LIGHT1 and LIGHT2

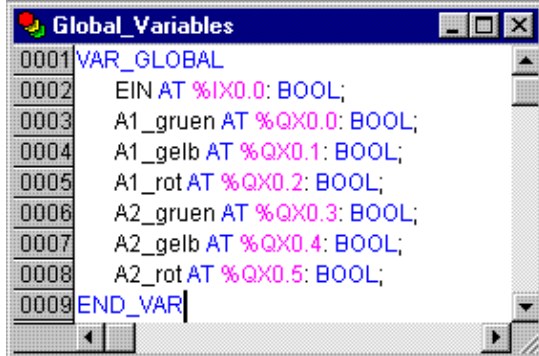


These deliver the Boolean value of each of the six lights to the above mentioned six outputs for each step of the block SEQUENCE. We are not, however, declaring the output variables which are foreseen within the PLC_PRG block but under Resources for Global Variables instead. The Boolean input variable IN, which is used to set the variable START in the block SEQUENCE to TRUE, can be set in the same way. ON is also allocated to an IEC address.

Select the tab Resources and open the list Global Variables.

Make the declaration as follows:

Declaration of the Input-/Output Variables for a CAN-Configuration



The name of the variable (e.g. IN) is followed, after AT, by a percent sign which begins the IEC address. I stands for input, Q for output, B (used in this example) stands for byte and the individual bits of the module are addressed using 0.0 (0.1, 0.2, etc.). We will not do the needed controller configuration here in this example, because it depends on which target package you have available on your computer. Please see PLC configuration for further information.

We now want to finish off the block PLC_PRG.

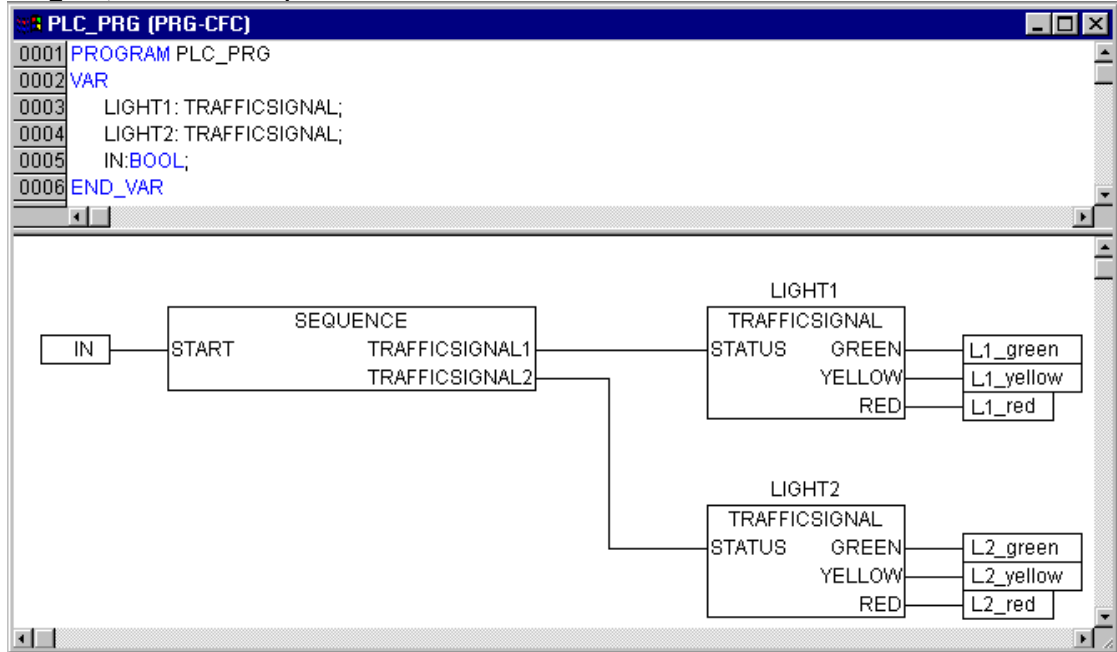
For this we go into the editor window. We have selected the Continuous Function Chart editor and we consequently obtain, under the menu bar, a CFC symbol bar with all of the available elements (see The Continuous Function Chart Editor).

Click on the right mouse key in the editor window and select the element **Box**. Click on the text AND and write "SEQUENCE" instead. This brings up the block SEQUENCE with all of the already defined input and output variables. Insert two further block elements which you name PHASES. Phases is a function block and this causes you to obtain three red question marks over the block which you replace with the already locally declared variables LIGHT1 and LIGHT2. Now set an element of the type **Input**, which award the title ON and six elements of the type **Output** which you award variable names to, as described, namely L1_green, L1_yellow, L1_red, L2_green, L2_yellow, L2_red.

All of the elements of the programme are now in place and you can connect the inputs and outputs, by clicking on the short line at the input/output of an element and dragging this with a constantly depressed mouse key to the input/output of the desired element.

Your program should finally look like shown in the following:

PLC_PRG, Declaration and presentation with the continuous function chart editor




TRAFFICSIGNAL simulation

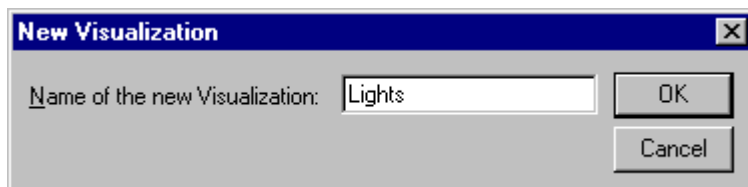
Now test your program in simulation mode. Compile ('Project' 'Build') and load ('Online' 'Login') it. Start the program by 'Online' 'Start', then set variable ON to TRUE, e.g. by a double-click on the entry "ON" in the input box of the CFC editor. This will mark the variable as prepared to be set to <TRUE>. Then press <Ctrl><F7> or command 'Online' 'Write values', to set the value. Now variable START in ABLAUF (which we had set to TRUE manually in the first extension level of the program) gets this value by variable ON, which is used in PLC_PRG. This will make run the traffic light cycles. PLC_PRG has changed to a monitoring window. Click twice on the plus sign in the declaration editor, the variable display drops down, and you can see the values of the individual variables.

3.2 Visualizing a Traffic Signal Unit...

With the visualization of **CoDeSys** you can quickly and easily bring project variables to life. We will now plot two traffic signals and an ON-Switch for our traffic light unit which will illustrate the switching process.

Creating a new visualization

In order to create a visualization you must first select the range of **Visualization** in the Object Organizer. First click on the lower edge of the window on the left side with the **POU** on the register card with this symbol  and the name **Visualization**. If you now choose the command 'Project' 'Object Add', then a dialog box opens.

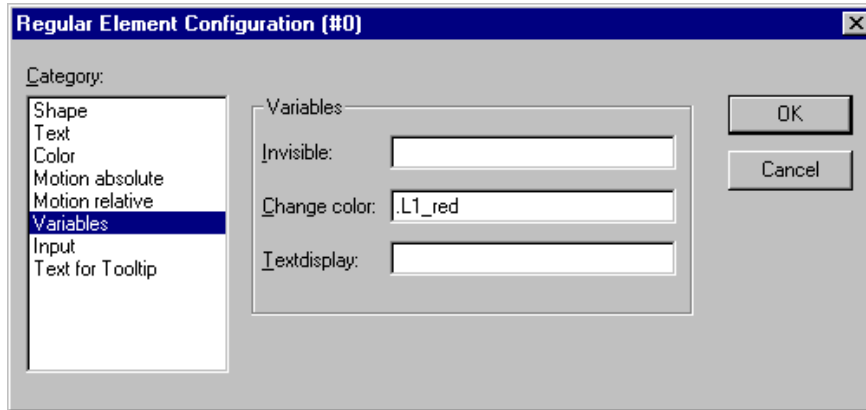


Enter here any name. When you confirm the dialog with **OK**, then a window opens in which you can set up your new visualization.

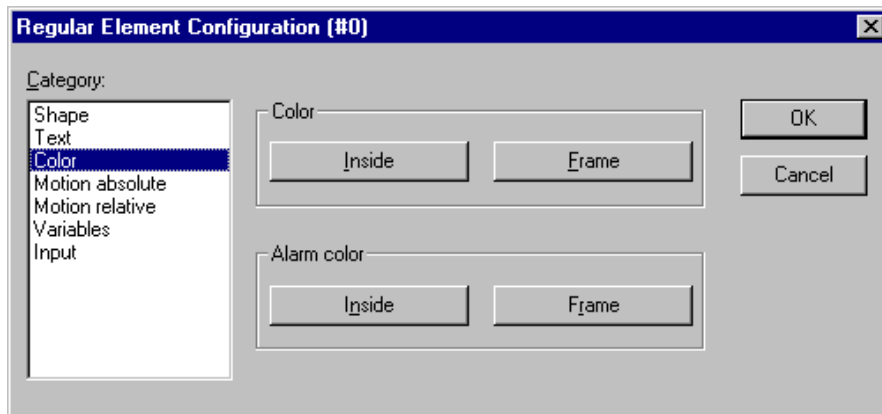
Insert element in Visualization

For our TRAFFICSIGNAL visualization you should proceed as follows:

- Give the command 'Insert' 'Ellipse' and try to draw a medium sized circle (?2cm). For this click in the editor field and draw with pressed left mouse button the circle in its length.
- Now double-click the circle. The dialog box for editing visualization elements opens
- Choose the category Variables and in the field Change color enter the variable name .L1_red or "L1_red". That means that the global variable L1_red will cause the color change as soon as it is set to TRUE. The dot before the variable name indicates that it is a global variable, but it is not mandatory.



- Then choose the category **Color** and click on the button **Inside** in the area **Color**. Choose as neutral a color as possible, such as black.
- Now click on the button **within** in the area **Alarm color** and choose the red which comes closest to that of a red light.



The resulting circle will normally be black, and when the variable RED from TRAFFICSIGNAL1 is TRUE, then its color will change to red. We have therefore created the first light of the first TRAFFICSIGNAL!

The other traffic lights

Now enter the commands 'Edit' 'Copy' (<Ctrl>+<C>) and then twice 'Edit' 'Paste' (<Ctrl>+<V>). That gives you two more circles of the exact same size lying on top of the first one. You can move the circles by clicking on the circle and dragging it with pressed left mouse button. The desired position should, in our case, be in a vertical row in the left half of the editor window. Double-click on one of the other two circles in order to open the configuration dialog box again. Enter in the field Change Color of the corresponding circle the following variables:

for the middle circle: L1_yellow

for the lowest circle: L1_green

Now choose for the circles in the category **Color** and in the area **Alarm color** the corresponding color (yellow or green).

The TRAFFICSIGNAL case

Now enter the command 'Insert' 'Rectangle', and insert in the same way as the circle a rectangle which encloses the three circles. Once again choose as neutral a color as possible for the rectangle and give the command 'Extras' 'Send to back' so that the circles are visible again.

If simulation mode is not yet turned on, you can activate it with the command 'Online' 'Simulation'.

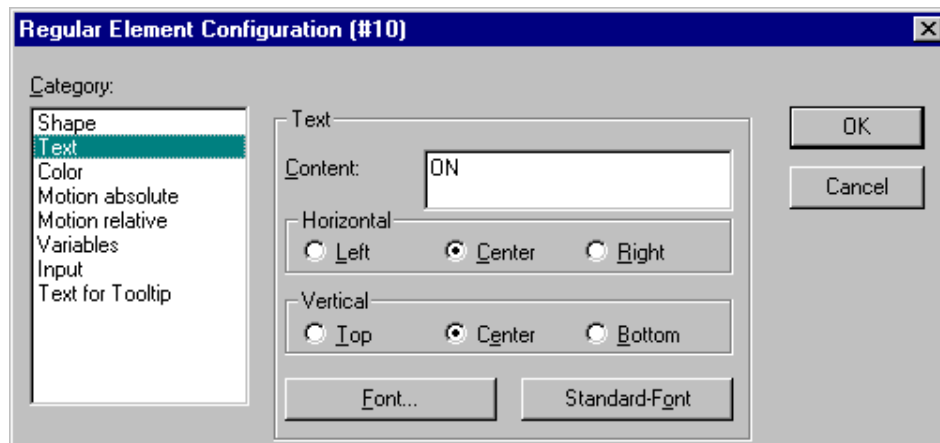
If you now start the simulation with the commands 'Online' 'Login' and 'Online' 'Run', then you can observe the color change of the first traffic signal.

The second traffic signal

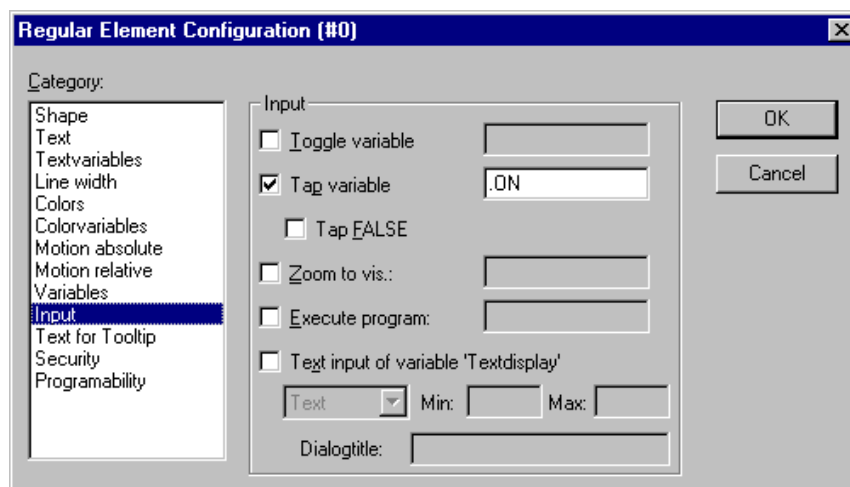
The simplest way to create the second traffic signal is to copy all of the elements of the first traffic signal. For this you select all elements of the first traffic signal and copy them (as before with the lights of the first traffic signal) with the commands 'Edit' 'Copy' and 'Edit' 'Paste'. You then only have to change the text "TRAFFICSIGNAL1" in the respective dialog boxes into "TRAFFICSIGNAL2", and the visualization of the second traffic signal is completed.

The ON switch

Insert a rectangle and award it, as described above, a colour for a traffic light of your choice and enter .ON at **Variables** for the **Change color**. Enter "ON" in the input field for **Content** in the category Text.



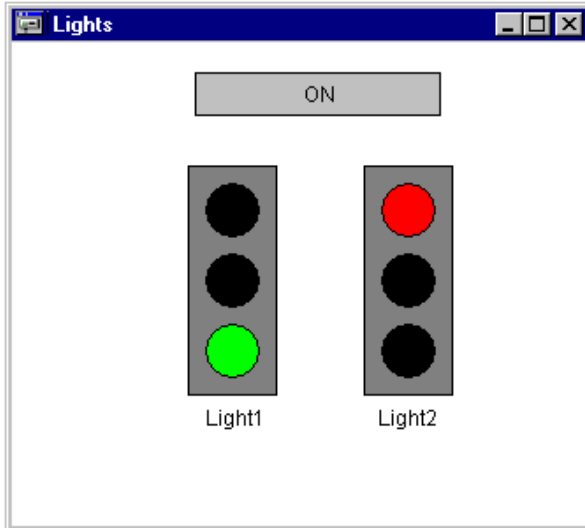
In order to set the variable ON to TRUE with a mouse click on the switch, activate option 'Toggle variable' in category 'Input' and enter variable name ".ON" there. Variable keying means that when a mouse click is made on the visualization element the variable .ON is set to the value TRUE but is reset to the value FALSE when the mousekey is released again (we have created hereby a simple switch-on device for our traffic lights program).



Font in the visualization

In order to complete the visualization you should first insert two more rectangles which you place underneath the traffic signals.

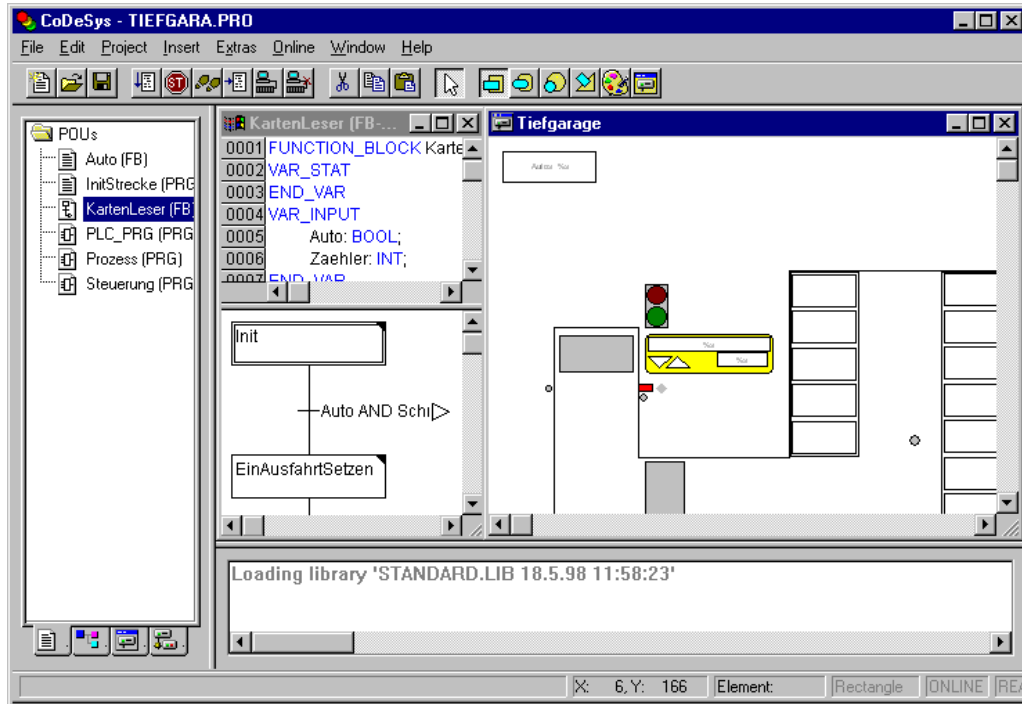
In the visualizations dialog box set white in the category **Color** for **Frame** and write in the category **Text** in the field **Contents** "Light1" or "Light2". Now your visualization looks like this:



4 The Individual Components

4.1 The Main Window...

Main window components



The following elements are found in the main window of **CoDeSys** (from top to bottom):

The Menu bar

The Tool bar (optional); with buttons for faster selection of menu commands.

The Object Organizer with register cards for POU's, Data types, Visualizations, and Resources

A vertical screen divider between the Object Organizer and the Work space of CoDeSys

The Work space in which the editor windows are located

The Message Window (optional)

The Status bar (optional); with information about the current status of the project

See also:

Context Menu

Menu bar

The menu bar is located at the upper edge of the main window. It contains all menu commands.



Tool bar

By clicking with the mouse on a symbol you can select a menu command more quickly. The choice of the available symbols automatically adapts itself to the active window.

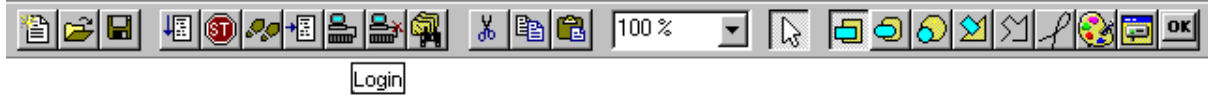
The command is only carried out when the mouse button is pressed on the symbol and then released.

The Main Window...





If you hold the mouse pointer for a short time on a symbol in the tool bar, then the name of the symbol is shown in a Tooltip.

In order to see a description of each symbol on the tool bar, select in Help the editor about which you want information and click on the tool bar symbol in which you are interested.

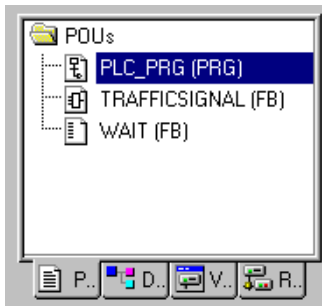
The display of the tool bar is optional (see 'Project' 'Options' category Desktop).



Object Organizer

The Object Organizer is always located on the left side of **CoDeSys**. At the bottom there are four register cards with symbols for the four types of objects  **POUs**,  **Data types**,  **Visualizations** and  **Resources**. In order to change between the respective object types click with the mouse on the corresponding register card or use the left or right arrow key.

You will learn in chapter Managing Objects in a Project how to work with the objects in the Object Organizer.



Screen divider

The screen divider is the border between two non-overlapping windows. In **CoDeSys** there are screen dividers between the Object Organizer and the Work space of the main window, between the interface (declaration part) and the implementation (instruction part) of POUs and between the Work space and the message window.

You can move the screen divider with the mouse pointer. You do this by moving the mouse with the left mouse button pressed.

Make sure the screen divider always remains at its absolute position, even when the window size has been changed. If it seems that the screen divider is no longer present, then simply enlarge your window.

Work space

The Work space is located on the right side of the main window in **CoDeSys**. All editors for objects and the library manager are opened in this area. The current object name appears in the title bar; in the case of POUs an abbreviation for the POU type and the programming language currently in use appears in brackets after it.

You find the description of the editors in the chapter The Editors

Under the menu item **'Window'** you find all commands for window management.

Message window

The message window is separated by a screen divider underneath the work space in the main window.

It contains all messages from the previous compilations, checks or comparisons. Search results and the cross-reference list can also be output here.

If you double-click with the mouse in the message window on a message or press <Enter>, the editor opens with the object. The relevant line of the object is selected. With the commands 'Edit' 'Next error' and 'Edit' 'Previous error' you can quickly jump between the error messages.

The display of the message window is optional (see 'Window' 'Messages').

Status bar

The status bar at the bottom of the window frame of the main window in **CoDeSys** gives you information about the current project and about menu commands.

If an item is relevant, then the concept appears on the right side of the status bar in black script, otherwise in gray script.

When you are working in online mode, the concept **Online** appears in black script. If you are working in the offline mode it appears in gray script.

In Online mode you can see from the status bar whether you are in the simulation (**SIM**), the program is being processed (**RUNS**), a breakpoint is set (**BP**), or variables are being forced (**FORCE**).

With text editor the line and column number of the current cursor position is indicated (e.g. **Line:5, Col.:11**). In online mode '**OV**' is indicated black in the status bar. Pressing the <Ins> key switches between Overwrite and Insert mode.

If the mouse point is in a visualization, the current **X** and **Y position** of the cursor in pixels relative to the upper left corner of the screen is given. If the mouse pointer is on an **Element**, or if an element is being processed, then its number is indicated. If you have an element to insert, then it also appears (e.g. **Rectangle**).

If you have chosen a menu command but haven't yet confirmed it, then a short description appears in the status bar.

The display of the statusbar is optional (see 'Project' 'Options' category Desktop).

Context Menu

Shortcut: <Shift>+<F10>

Instead of using the menu bar for executing a command, you can use the right mouse button. The menu which then appears contains the most frequently used commands for a selected object or for the active editor. The choice of the available commands adapts itself automatically to the active window. The choice of the available commands adapts itself automatically to the active window.

4.2 Project Options...

'Project' 'Options'

With this command the dialog box for setting options is opened. The options are divided into different categories. Choose the desired category on the left side of the dialog box by means of a mouse click or using the arrow keys and change the options on the right side.

An image of the options which are set for the current project, will be found in the Resources tab in component 'Workspace'.

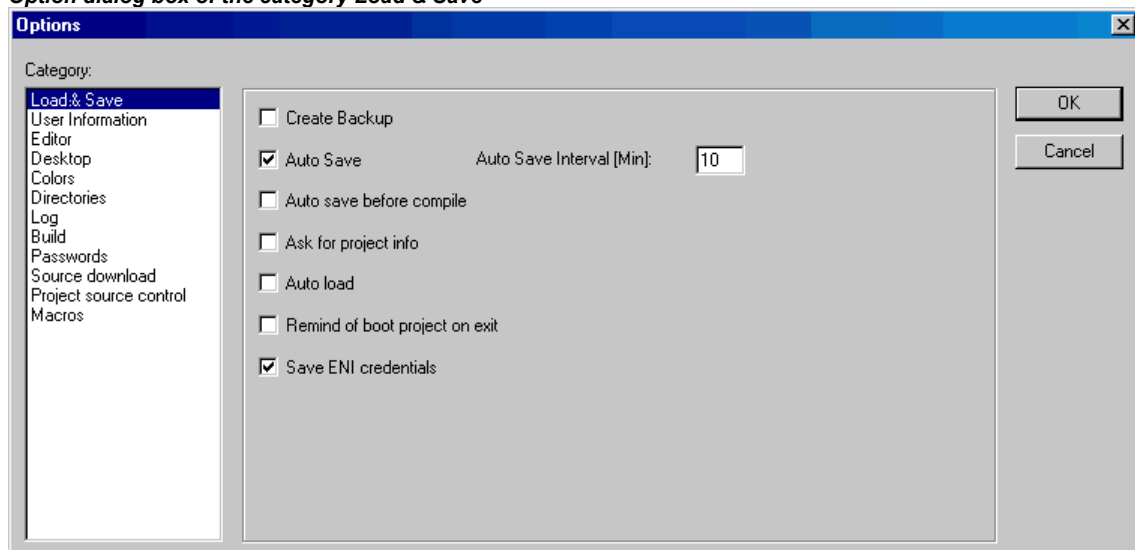
The settings amongst other things serve to configure the view of the main window. They are, unless determined otherwise, saved in the file "**CoDeSys.ini**" and restored at the next **CoDeSys** startup.

The categories:	stored in CoDeSys	stored in project
Load & Save	X	
User information	X	
Editor	X	
Desktop	X	
Color	X	
Directories	Cat. Common	Cat. Project
Log	X	
Build		X
Passwords		X
Source download	X	
Symbol configuration	X	
Project source control		X
Macros		X

Options for Load & Save

If you choose this category in the Options dialog box , then you get the following dialog box:

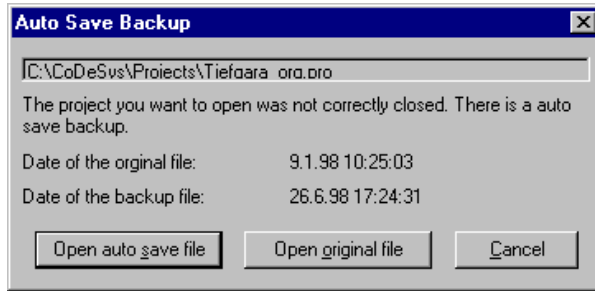
Option dialog box of the category Load & Save



When activating an option, a check (✓) appears before the option.

Create Backup: CoDeSys creates a backup file at every save with the extension ".bak". Contrary to the *.asd-file (see below, 'Auto Save') this *.bak-file is kept after closing the project. So you can restore the version you had before the last project save.

Auto Save: While you are working, your project is saved according to a defined time interval (**Auto Save Interval**) to a temporary file with the extension ".asd" in the projects directory. This file is erased at a normal exit from the program. If for any reason CoDeSys is not shut down "normally" (e.g. due to a power failure), then the file will not get erased. When you open the file again the following message appears:



You can now decide whether you want to open the original file or the auto save file.

If a library *.lib is opened as project, a corresponding auto-save-file "*.asl" will be created.

Auto save before compile: The project will be saved before each compilation. In doing so a file with the extension ".asd" resp. ".asl" will be created, which behaves like described above for the option 'Auto Save'.

Ask for project info: When saving a new project or saving a project under a new name, the project info is automatically called. You can visualize the project info with the command 'Project' 'Project info' and also process it.

Auto Load: At the next start of CoDeSys the last open project is automatically loaded. The loading of a project at the start of CoDeSys can also take place by entering the project in the command line.

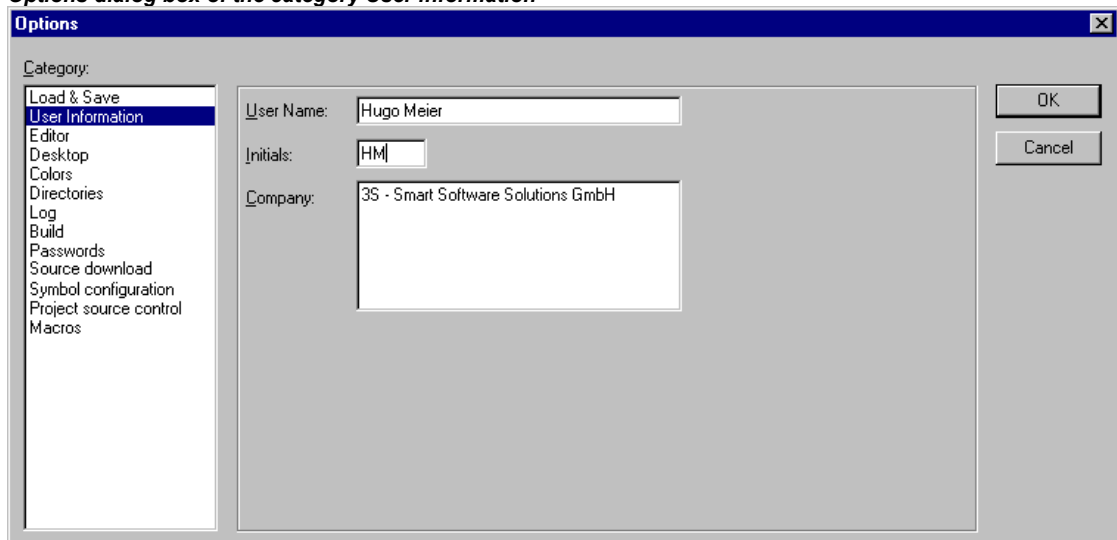
Remind of boot project on exit: If the project has been modified and downloaded without creating a new boot project since the last download of a boot project, then a dialog will advise the user before leaving the project: "No boot project created since last download. Exit anyway ?".

Save ENI credentials: User name and Password, as they might be inserted in the Login dialog for the ENI data base, will be saved. Concerning the access data, entered once by the user at 'Open project from source code manager' (see chapter 4.2, 'File' 'Open') in this case additionally user name and password will be saved in the codesys.ini file.

Options for User information

If you choose this category in the Options dialog box, then you get the following dialog box:

Options dialog box of the category User information



To User information belong the **Name** of the user, his **Initials** and the **Company** for which he works. Each of the entries can be modified. The settings will be applied to any further projects which will be created with **CoDeSys** on the local computer.

Options for Editor

If you choose this category in the Options dialog box, then you get the dialog box shown below.

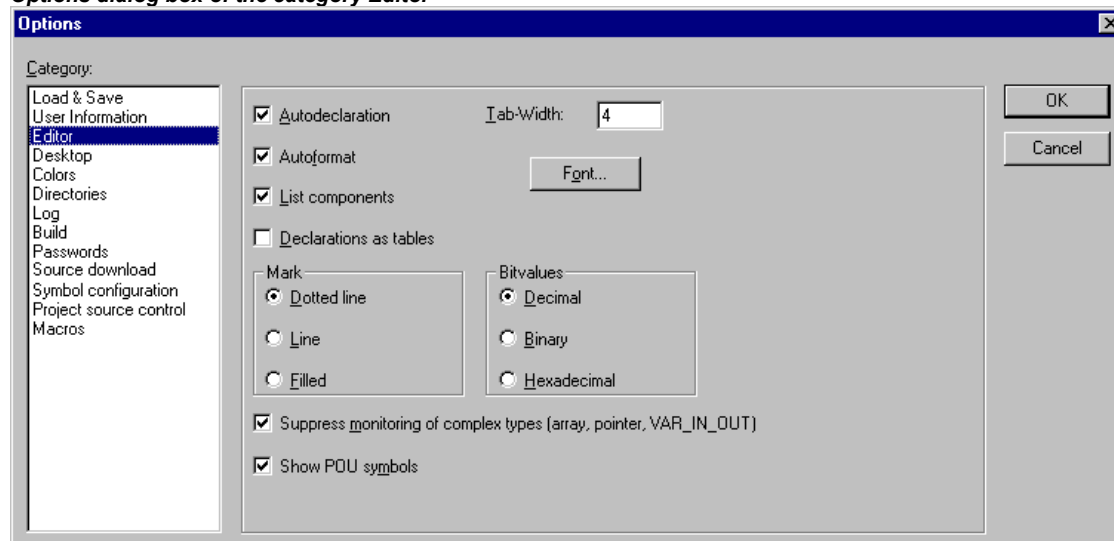
You can make the following settings for the Editors:

Autodeclaration: If this option is activated, then after the input of a not-yet-declared variable a dialog box will appear in all editors with which this variable can be declared.

Autoformat: If this option is activated, then CoDeSys executes automatic formatting in the IL editor and in the declaration editor. When you finish with a line, the following formatting is made: 1. Operators written in small letters are shown in capitals; 2. Tabs are inserted so that the columns are uniformly divided.

List components: If this option is activated, then the "Intellisense functionality" will be available to work as an input assistant. This means that if you insert a dot at a position where an identifier should be inserted, then a selection list will open, offering all global variables which are found in the project. If you insert the name of a function block instance, then you will get a selection list of all inputs and outputs of the instanced function block. The "Intellisense function" is available in editors, in the Watch- and Receiptmanager, in visualizations and in the Sampling Trace.

Options dialog box of the category Editor



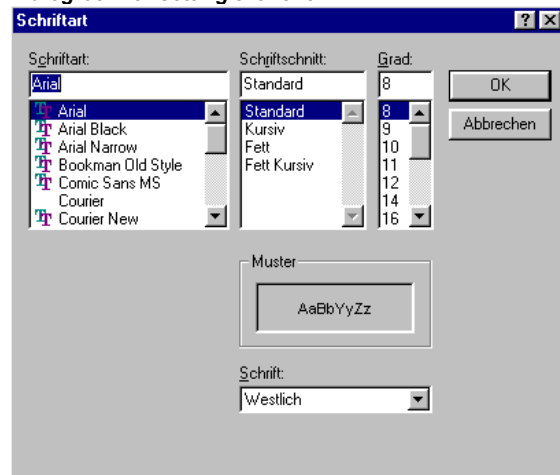
Declarations as tables: If this option is activated, then you can edit variables in a table instead of using the usual declaration editor. This table is sorted like a card box, where you find tabs for input variables, output variables, local variables, and in_out variables. For each variable there are edit fields to insert **Name**, **Address**, **Type**, **Initial** and **Comment**.

Tab-Width: In the field **Tab-Width** in the category Editor of the Options dialog box you can determine the width of a tab as shown in the editors. The default setting is four characters, whereby the character width depends upon the font which is chosen.

Font: By clicking on the button **Font** in the category Editor of the Options dialog box you can choose the font in all CoDeSys editors. The font size is the basic unit for all drawing operations. The choice of a larger font size thus enlarges the printout, even with each editor of CoDeSys.

After you have entered the command, the font dialog box opens for choosing the font, style, and font size.

Dialog box for setting the font

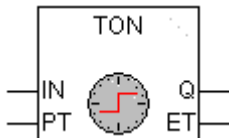


Mark: When choosing Mark in the Editor category in the Options dialog box you can choose whether the current selection in your graphic editors should be represented by a dotted rectangle (Dotted), a rectangle with continuous lines (Line) or by a filled-in rectangle (Filled). In the last case the selection is shown inverted.

Bitvalues: When choosing Bitvalues in the category Editor of the Options dialog box you can choose whether binary data (type BYTE, WORD, DWORD) during monitoring should be shown Decimal, Hexadecimal, or Binary.

Suppress monitoring of complex types (Array, Pointer, VAR_IN_OUT): If this option is activated, complex data types like arrays, pointers, VAR_IN_OUTs will not get displayed in the monitoring window in online mode.

Show POU symbols: If this option is activated, in the module boxes which are inserted to a graphic editor, additionally symbols will get displayed, if those are available in the library folder as bitmaps. The name of the bitmap-file must be composed of the name of the module and the extension ".bmp". Example: For module TON there is a symbol file TON.bmp available. The box will be displayed as follows:



Options for the Desktop

If you choose this category in the Options dialog box, then you get the dialog box shown below.

When an option is activated, a check appears in front of it.

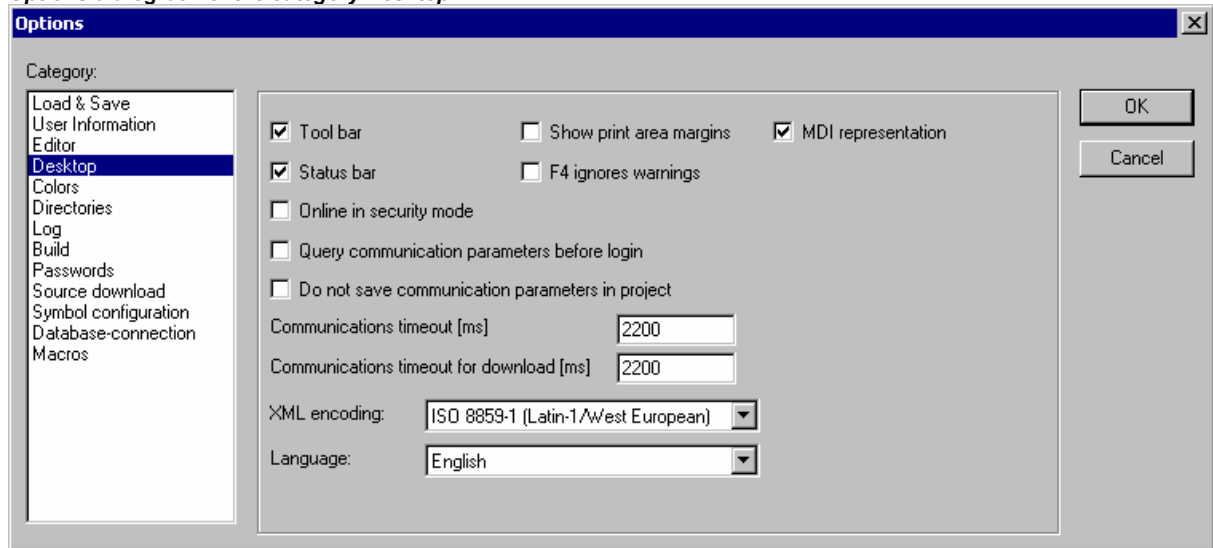
Tool bar: The tool bar with the buttons for faster selection of menu commands becomes visible underneath the menu bar.

Status bar: The status bar at the lower edge of the CoDeSys main window becomes visible.

Online in Security mode: In Online mode with the commands 'Run', 'Stop', 'Reset', 'Toggle Breakpoint', 'Single cycle', 'Write values', 'Force values' and 'Release force', a dialog box appears with the confirmation request whether the command should really be executed. If supported by the target system, an extended dialog might be available when you want to load the actual project from the programming system to the PLC. If there is already a project on the PLC, this dialog will display the project information of that project as well as the information of the project currently to be loaded. This project information also will be available in case of creating a boot project when there is already one on the PLC.

This option is saved with the project.

Options dialog box of the category Desktop



Query communication parameters before login: As soon as the command 'Online' 'Login' is executed, first the communication parameters dialog will open. To get in online mode you must first close this dialog with OK.

Do not save communication parameters in project: The settings of the communication parameters dialog ('Online' 'Communication Parameters') will not be saved with the project.

Show print area margins: In every editor window, the limits of the currently set print range are marked with red dashed lines. Their size depends on the printer characteristics (paper size, orientation) and on the size of the "Content" field of the set print layout (menu: 'File' "Documentation Settings").

F4 ignores warnings: After compilation, when F4 is pressed in a message window, the focus jumps only to lines with error messages; warning messages are ignored.

MDI representation: Per default this option (**M**ultiple-**D**ocument-**I**nterface) is activated and thus several windows can be opened at the same time. If the option is deactivated (SDI mode) only one window can be opened and will be displayed in full screen mode. Exception: The action of a program and the program itself can be displayed side by side even in MDI mode.

Communications timeout [ms]: for standard communication services: Time span in milliseconds, after which the communication to the target system will be terminated if no more activity is detected. Possible values: 1-10000000 ms.

Communications timeout for download [ms]: for long lasting communication services (program download, file up- and download, boot project creation and check): Time span in milliseconds after which the communication to the target system will be terminated if no more activity is detected (Download Wait Time). Possible values: 1-10000000 ms.

XML-Encoding: The format for XML-exports can be selected. The default setting is "ISO 8859-1". This concerns the communication via ENI, Message Interface and COM Automation Interface, as well as each user-triggered XML-export from CoDeSys. An exception is the XML-export of the Licensing Manager.

Language: Define here, in which language the menu and dialog texts should be displayed.

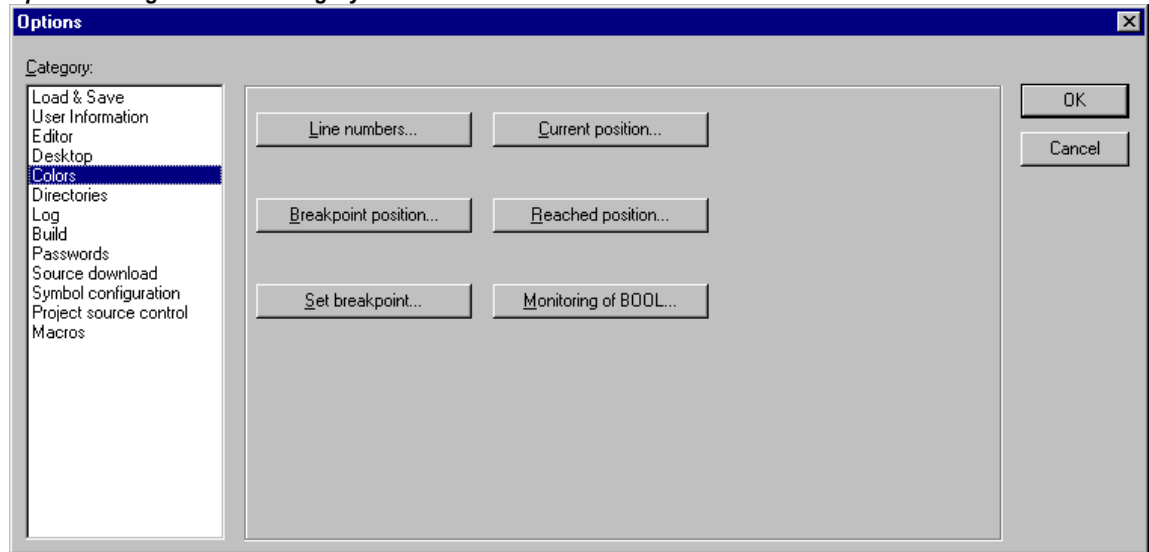
Note: Please note, that the language choice is not possible under Windows 98T !

Note: The settings will be stored in CoDeSys.

Options for Colors

If you choose this category in the Options dialog box , then you get the following dialog box:

Options dialog box of the category Color



You can edit the default color setting of **CoDeSys**. You can choose whether you want to change the color settings for **Line numbers** (default presetting: light gray), for **Breakpoint positions** (dark gray), for a **Set breakpoint** (light blue), for the **Current position** (red), for the **Reached Positions** (green) or for the **Monitoring of Boolean values** (blue).

If you have chosen one of the indicated buttons, the dialog box for the input of colors opens.

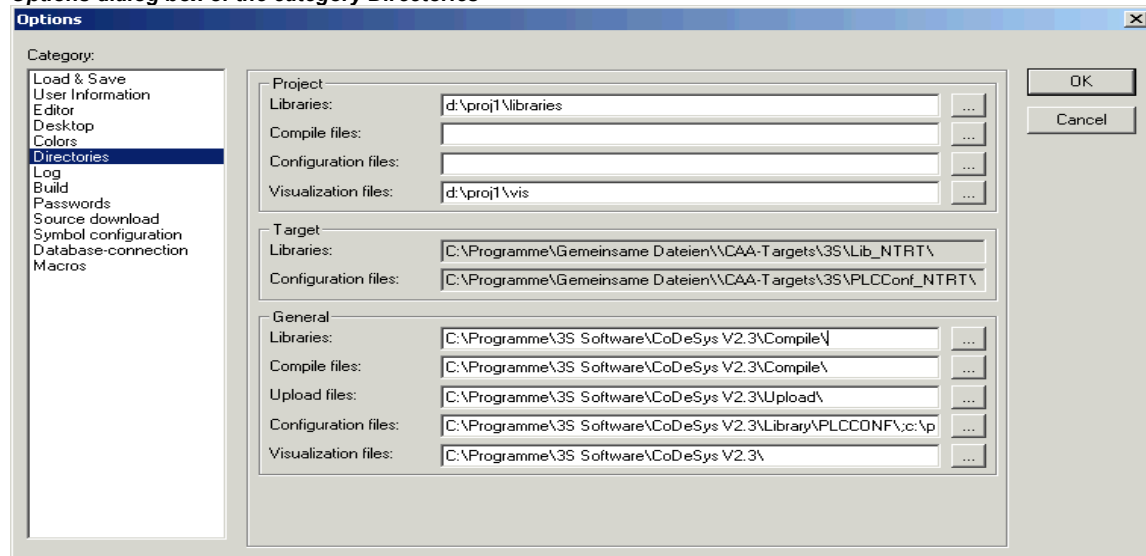
Dialog box for setting colors



Options for Directories

If you choose this category in the Options dialog box, then you get the following dialog box:

Options dialog box of the category Directories



Directories can be entered in the **Project** and **Common** areas for CoDeSys to use in searching for **Libraries**, controller **configuration** and **Visualization files**, as well as for storing **Compile** and source-**Upload files**. (Regard: Compile files for example are map- and list-files, not however e.g. symbol files ! The latter will be saved in the project directory.) If you activate the button (...) behind a field, the directory selection dialog opens. For library and configuration files, several paths can be entered for each, separated by semicolons ";".

Please regard:

- Library paths can be entered based on the project file's path by prefixing a dot ".". For example: If you enter ".libs", libraries will also be searched in 'C:\programs\projects\libs', if the current project is in 'C:\programs\projects'. For information on library paths see also: 'Chapter 6.4, 'Insert' 'Additional Library'.
- Do not use empty spaces and special characters except for "_" in the directory paths.

Information in the **Project** area is stored with the project.

Information in the **Common** area is written to the ini-file of the programming system and thus applies to all projects.

The **Target** area just displays the directories for libraries and configuration files set in the target system, e.g. through entries in the Target file. These fields cannot be edited, but an entry can be selected and copied (right mouse button context menu).


CoDeSys generally searches first in the directories entered in 'Project', then in those in 'Target' (defined in the Target file), and finally those listed under 'Common'. If two files with the identical names are found, the one in the directory that is browsed first will be used.

Options for Log

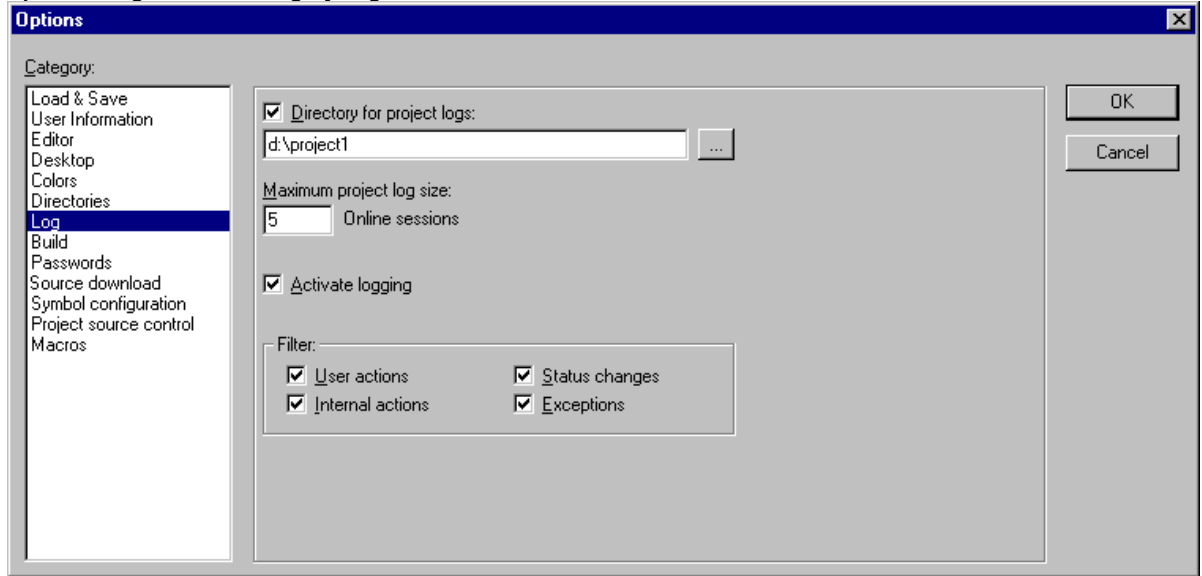
If you choose this category in the Options dialog box, then you get the dialog box shown below.

In this dialog, you can configure a file that acts as a project log, recording all user actions and internal processes during Online mode processing (see also: Log).

If an existing project is opened for which no log has yet been generated, a dialog box opens which calls attention to the fact that a log is now being set up that will receive its first input after the next login process.

The log is automatically stored as a binary file in the project directory when the project is saved. If you prefer a different target directory, you can activate the option **Directory for project logs:** and enter the appropriate path in the edit field. Use the  button to access the "Select Directory" dialog for this purpose.

Option dialogue for the category Log



The log file is automatically assigned the name of the project with the extension .log. The maximum number of **Online sessions** to be recorded is determined by **Maximum project log size**. If this number is exceeded while recording, the oldest entry is deleted to make room for the newest.

The Log function can be switched on or off in the Option field **Activate logging**.

You can select in the **Filter** area which actions are to be recorded: User actions, Internal actions, Status changes, Exceptions. Only actions belonging to categories checked here will appear in the Log window and be written to the Log file. (For a description of the categories, please see Log).

The Log window can be opened with the command 'Window' 'Log'.

Options for Build

If you choose this category in the Options dialog box , then you get the dialog box shown below.

Debugging: It depends on the target descriptions whether this option can be activated/deactivated by the user resp. which is the given setting . If it is activated, additional debugging code is created, that is the code can become considerably larger. The debugging code is needed in order to make use of the debugging functions offered by CoDeSys (e.g. breakpoints). When you switch off this option, project processing becomes faster and the size of the code decreases. The option is stored with the project.

Replace constants: The value of each constant of scalar type (thus not for strings, arrays and structures) is loaded directly, and in Online mode the constants are displayed in green. Forcing, writing and monitoring of a constant is then no longer possible. If the option is deactivated, the value is loaded into a storage location via variable access (this does in fact allow writing the variable value, but implies longer processing time).

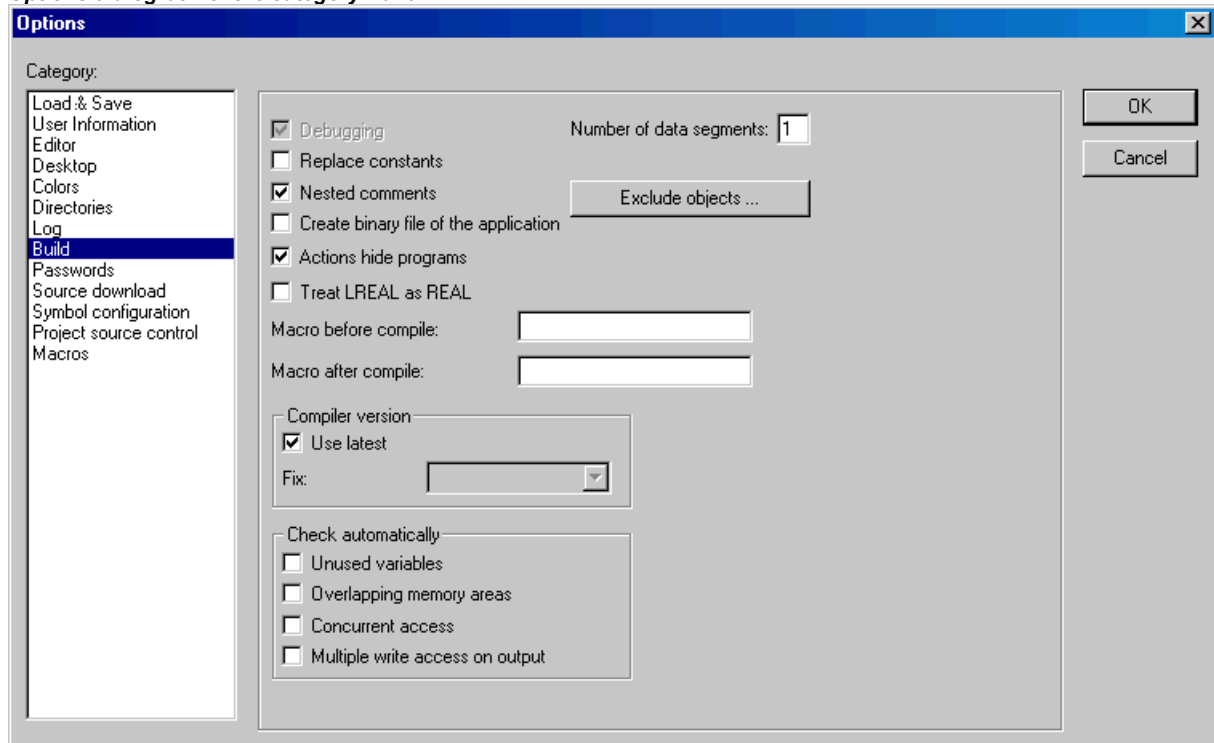
Nested comments: Comments can be placed within other comments. Example:

```
(*
a:=inst.out; (* to be checked *)
b:=b+1;
*)
```

Here the comment that begins with the first bracket is not closed by the bracket following "checked," but only by the last bracket.

Attention: Currently this option must be used carefully: If the setting in the project does not match the setting chosen in a library which also has been created in CoDeSys and now is used is the project, compiler errors will occur, which are hard to interpret by the user and often cannot be cleared!

Options dialog box of the category Build



Create binary file of the application: A binary image of the generated code (boot project) is created in the project directory during compilation. File name: <project_name>.bin. By comparison, the command 'Online' 'Create boot project' sets up the boot project on the controller.

Actions hide programs: This option is activated per default, when a new project is created. It means: If a local action has the same name like a global variable or a program, the following hierarchy is valid: local variable before local action before global variable before program.

Regard: If an existing project is opened, which has been created with a previous CoDeSys version, the option will be deactivated per default. Thus the previously valid hierarchy (local variable before global variable before program before local action) can be kept.

Treat LREAL as REAL: If this option is activated, (availability depends on runtime system; default: not activated), the compile will handle LREAL values as REAL values. This can be used for creating platform independent projects.

Number of data segments: Here you define how many memory segments should be allocated in the PLC for the project data. This space is needed to make possible Online Change even if new variables have been added. If during compilation you get the message "Out of global data memory...", enter a higher number. In this regard local program variables will be handled like global variables.

Exclude objects: This button opens the dialog **Exclude objects from build**: In the tree of project components select those POUs which should not be regarded during compilation and activate option **Exclude**. Hereupon the excluded POUs will be displayed green-colored in the selection tree. Press button **Exclude unused**, if you just want to get displayed those POUs which are currently used in the program. Regard that a single object which is selected in the Object Organizer can also be excluded from build by using the command 'Exclude from build' from the context menu.

Compiler Version: Here you define the compiler version to be used. CoDeSys versions after V2.3.3 (version, service pack, patch) will include besides the actual compiler version also the previous compiler versions (back to V2.3.3). If you always want to get the project compiled with the latest compiler version, activate option **Use latest**. In this case however it will be checked whether the currently opened programming system is also of that version. If this is not true, the compiler version matching the actually used programming system version will be used! If the project should be compiled with a specific version, define this via the selection list at **Fix**.

In order to exert control over the compilation process you can set up two macros:

The macro in the **Macro before compile** field is executed before the compilation process; the macro in the **Macro after compile** field afterwards. The following macro commands can not, however, be used here: file new, file open, file close, file save as, file quit, online, project compile, project check, project build, project clean, project rebuild, debug, watchlist.

Check automatically:

In order to get the semantic correctness checked at each compilation of the project the following options can be activated:

Unused variables

Overlapping memory areas

Concurrent access

Multiple write access on output

The results will be displayed in the message window. These checks also can be initiated by the respective commands of the 'Check' submenu in the 'Project' menu.

If supported by the target system, negative check results will produce compiler errors.

Note: All entries in the Build Options dialog are stored with the project.

Passwords

To protect your files from unauthorized access CoDeSys offers the option of using a password to protect against your files being opened or changed.

If you choose this category in the Options dialog box, then you get the corresponding dialog box.

Enter the desired password in the field **Password**. For each typed character an asterisk (*) appears in the field. You must repeat the same word in the field **Confirm Password**. Close the dialog box with **OK**. If you get the message:

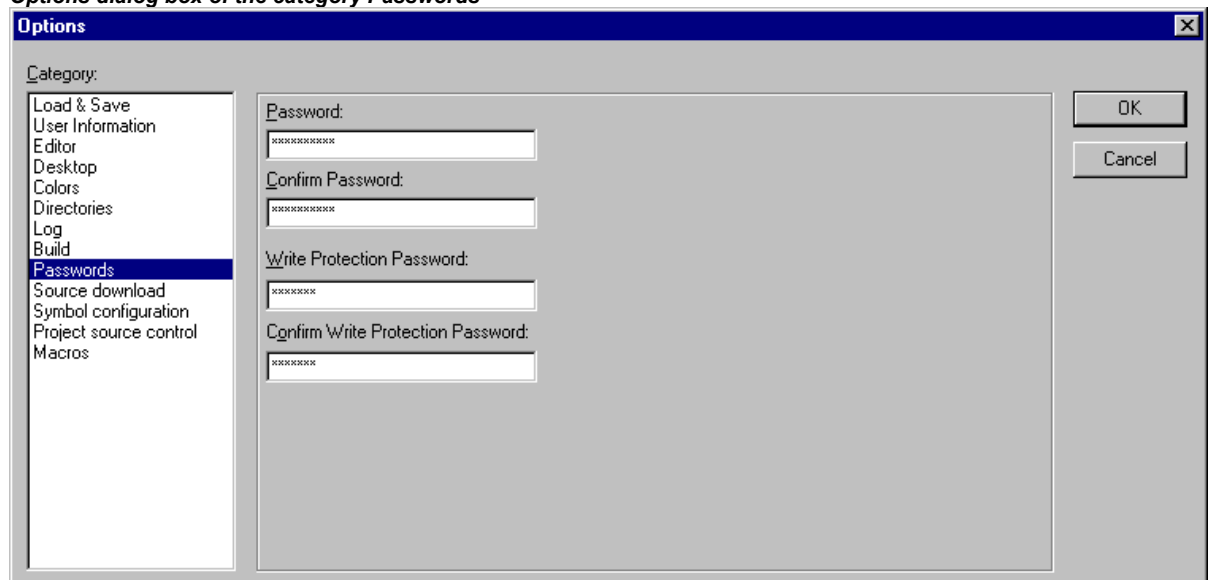
"The password does not agree with the confirmation",

then you made a typing error during one of the two entries. In this case repeat both entries until the dialog box closes without a message.

If you now save the file and then reopen it, then you get a dialog box in which you are requested to enter the password. The project can then only be opened if you enter the correct password. Otherwise CoDeSys reports:

"The password is not correct."

Options dialog box of the category Passwords



Along with the opening of the file, you can also use a password to protect against the file being changed. For this you must enter a password in the field **Write Protection Password** and confirm this entry in the field underneath.

A write-protected project can be opened without a password. For this simply press the button **Cancel**, if CoDeSys tells you to enter the write-protection password when opening a file. Now you can compile the project, load it into the PLC, simulate, etc., but you cannot change it.

Of course it is important that you memorize both passwords. However, if you should ever forget a password, then contact the manufacturer of your PLC.

The passwords are saved with the project.

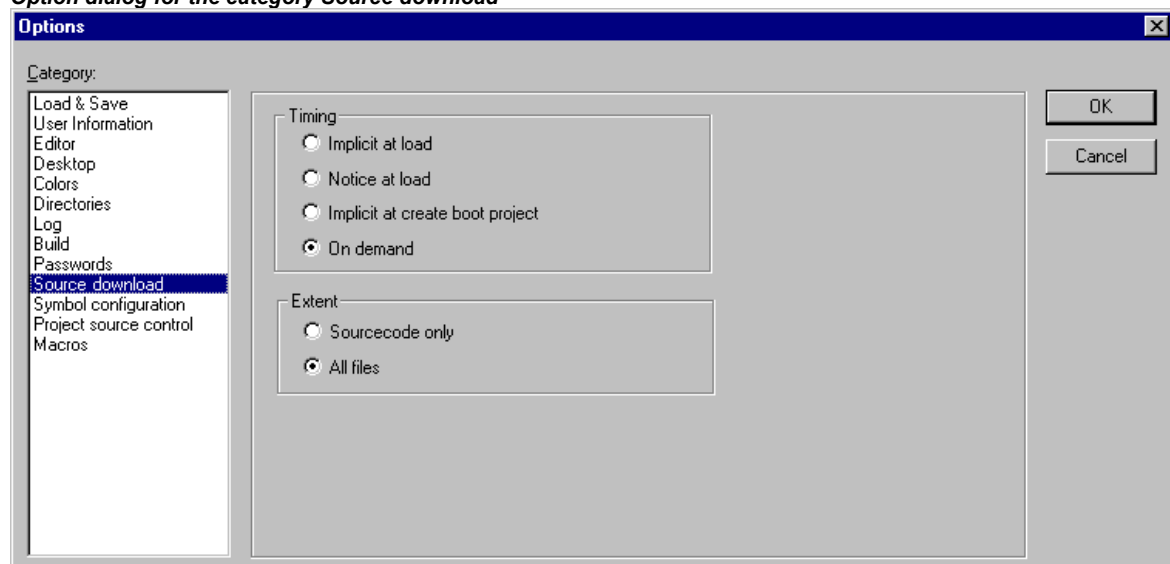
In order to create differentiated access rights you can define user groups and "Passwords for user groups").

Additionally regard the extended possibilities to protect a project by encryption (see Chapter 4.3, 'File', 'Save as'), which for example can help to protect a library from getting used without having entered a key.

Source download

The following dialog will be opened when you select this category:

Option dialog for the category Source download



You can choose to which **Timing** and what **Extent** the project is loaded into the controller system. The option **Sourcecode only** exclusively involves just the CoDeSys file (file extension .pro). The option **All files** also includes files such as the associated library files, visualization bitmaps, configuration files, etc.

Using the option **Implicit at load** allows the selected file range to be automatically loaded into the controller system on the command '**Online**' 'Download'.

Using the option **Implicit at create boot project** allows the selected file range to be automatically loaded into the controller system on the command 'Online' 'Create boot project'.

Using the option **Notice at load** offers a dialog, when the command '**Online**' 'Download' is given, with the question "Do you want to write the source code into the controller system?". Pressing **Yes** will automatically load the selected range of files into the controller system, or you can alternatively finish with **No**.

When using the option **On demand** the selected range of files must be expressly loaded into the controller system by giving the command 'Online' 'Sourcecode download'.

The project which is stored in the controller system can be retrieved by using 'File' 'Open' with Open project from PLC. The files will be unpacked in the process.

Options for 'Symbol Configuration'

The dialog presented here (not available in simulation mode !) is used for configuring the symbol file which will be created during each compilation of the project. The symbol file is created as a text file <project name>.sym respectively as a binary file <project name>.sdb (the format is depending on the used gateway version) in the project directory. The file is needed for data exchange with the controller via the symbolic interface and will be used for that purpose e.g. by OPC- or GatewayDDE-Server.

If the option Create symbol entries is activated, then symbol entries for the project variables will be automatically written to the symbol file. Otherwise only version info about file and project are contained.

If additionally the option **Dump XML symbol table** is activated, then also an XML file containing the symbol information will be created in the project directory. It will be named <project name>.SYM_XML.

Regard the following when configuring the symbol entries:

If option 'Symbol config from INI-file' is activated in the target settings, then the symbol configuration will be read from the codesys.ini file or from another ini-file which is defined there. (In this case the dialog 'Set object attributes' in CoDeSys cannot be edited).

If option 'Symbol config from INI-file' is not activated, the symbol entries will be generated in accordance with the settings you can make in the 'Set object attributes' dialog. You get there using the **Configure symbol file** button:

Dialog 'Set object attributes' (in option category Symbol configuration)



Use the tree-structured selection editor to mark the variables which should be entered in the symbol file. For this purpose you can select a POU's entry (e.g. Global Variables) which automatically will mark all variables belonging to this POU, or you can select single variables which you find listed for each POU in the tree. Then set the desired options in the lower part of the dialog box by clicking the mouse on the corresponding small boxes. Activated options are checked. The following options can be set:

Export variables of object: The variables of the selected object are exported in the symbol file.

The following options can take effect only if the **Export variables of object** option is activated:

Export data entries: Entries for access to the global variables are created for object's structures and arrays.

Export structure components: An individual entry is created for each variable component of object's structures.

Export array entries: An individual entry is created for each variable component of object's arrays.

Write Access: Object's variables may be changed by the OPC server.

Once the option settings for the currently selected variables are complete, other POU's can be also selected - without closing the dialog before - and given an option configuration. This can be carried out for any desired number of POU selections, one after the other. When the dialog box is closed by selecting **OK**, all configurations carried out since the dialog box was opened are applied.

Note: Regard the possibility of using pragmas in the declaration of a variable, which might define that the variable is taken to the symbol file with restricted access or that it is excluded from the symbol file.

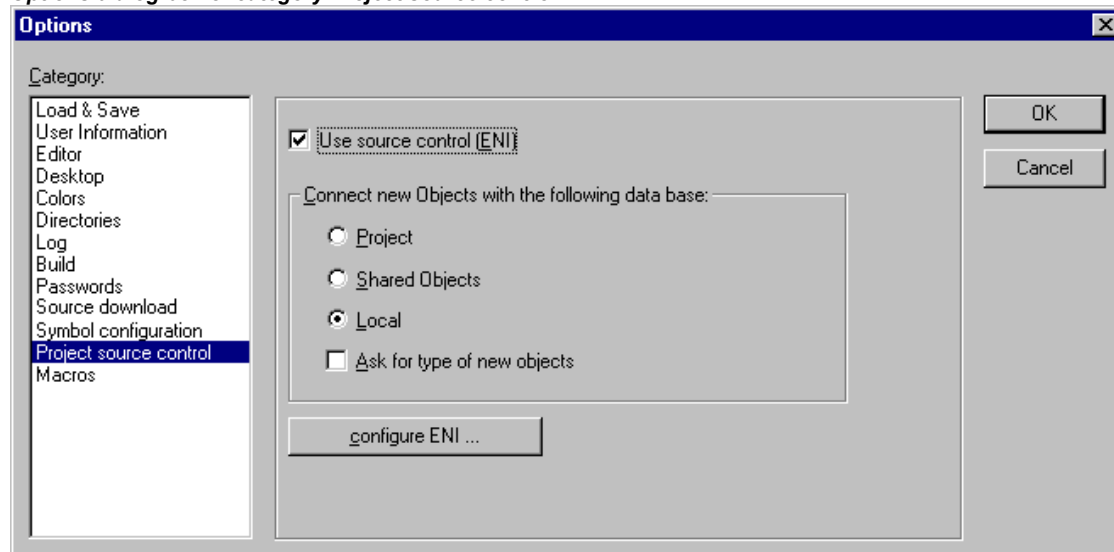
Options for 'Project source control'

This dialog is used to define whether the project should be managed in a project data base and to configure the ENI interface correspondingly.

Use source control (ENI): Activate this option, if you want to access a project data base via the ENI Server in order to administer all or a selection of POU's of the project in this data base. Preconditions: ENI Server and data base must be installed and you must be registered as an user in the database. See also the documentation for the ENI-Server resp. in chapter 'The CoDeSys ENI'.

If the option is activated, then the data base functions (Check in, Get last version etc.) will be available for handling the project POU's. Then some of the data base functions will run automatically like defined in the options dialogs, and in the menu 'Project' 'Data Base Link' you will find the commands for calling the functions explicitly. Besides that a tab 'Data base-connection' will be added in the dialog Properties, where you can assign a POU to a particular data base category.

Options dialog box of category Project source control



Connect new Objects with the following data base:

Here you set a default: If a new object is inserted in the project ('Project' 'Object' 'Add'), then it will automatically get assigned to that object category which is defined here. This assignment will be displayed in the object properties dialog ('Project' 'Object' 'Properties') and can be modified there later. The possible assignments:

Project: The POU will be stored in that data base folder which is defined in the dialog ENI configuration/Project objects in the field 'Project name'.

Shared Objects: The POU will be stored in that data base folder which is defined in the dialog ENI configuration/Shared objects in the field 'Project name'.

Local: The POU will not be managed in a ENI data base, but only will be stored locally in the project.

Besides 'Project objects' and 'Shared objects' there is a third data base category 'Compile files' for such objects which are not created until the project has been compiled. Therefore this category is not relevant for the current settings.

Ask for type of new objects: If this option is activated, then whenever a new object is added to the project, the dialog 'Object' 'Properties' will open, where you can choose to which of the three object categories mentioned above the POU should be assigned. By doing so the standard setting can be overwritten.

configure ENI: This button opens the first of three ENI configuration dialogs:

Each object of a project, which is determined to get managed in the ENI data base, can be assigned to one of the following data base categories: 'Project objects', 'Shared objects' or 'Compile files'. For each of these categories a separate dialog is available to define in which data base folder it should be stored and which presettings should be effective for certain data base functions:

Dialog ENI configuration / Project objects

Dialog ENI configuration / Shared objects

Dialog ENI configuration / Compile files

Note: Each object will be stored also locally (with project) in any case.

The dialog will open one after the other if you are doing a primary configuration. In this case a **Wizard** (Button **Next**) will guide you and the settings entered in the first dialog will be automatically copied to the other ones, so that you just have to modify them if you need different parameter values.

If you want to modify an existing configuration, then the three dialogs are combined in one window (three tabs).

If you have not yet logged in successfully to the data base before, then the Login dialog will be opened automatically.

Options for project objects and shared objects regarding the project data base

These dialogs are part of the configuration of the project data base options ('Project' 'Options' 'Project source control'). Here you define the access parameters for the data base categories 'Project objects' und 'Shared objects'. Both dialogs contain the same items. (A third dialog is available for the configuration of the access to the data base category 'Compile files', see next chapter)

ENI-Connection

TCP/IP-Address:	Address of the computer where the ENI-Server is running
Port:	Default: 80; must be the same as set in the configuration parameters of the ENI Server
Project name:	Name of the data base folder where the objects of this category should be stored. Press button ... to open a folder tree of the already existing data base projects. If the desired folder already exists, you can select it in this tree and its name will be entered in the 'Project name' edit field. If you had not logged in to the ENI Server until you try to open the folder tree by button ..., then you will first get the Login dialog where you must enter 'User name' and 'Password' as defined in your ENI user account to get access to the three data base categories.
Read only	If this option is activated, then only read access is possible to the above defined data base folder.

Dialog 'Project objects' in options category Project source control

Project objects

ENI-Connection

TCP/IP-Address: localhost

Port: 80

Project name:

Read only

Get latest Version

At Project Open with Query

Immediately after Changes in ENI with Query

Before any Compile with Query

Check out

Immediately at start of editing with Query

Check in

At Project Save with Query

After successful compile with Query

< Back Next > Cancel

Get latest Version

The data base function 'Get latest Version' (Menu 'Project' 'Data Base Link') copies the latest version of POUs from the above defined data base folder to the currently opened project, whereby the local version of objects will be overwritten. This will be done automatically for all objects, for which the version found in the data base differs from that in the project, as soon as one of the set timing conditions will meet. Activate the desired time options by setting a check mark:

At Project Open	As soon as the project is opened in CoDeSys
Immediately after Changes in ENI	As soon as a newer version of the POU is checked in to the data base (e.g. by another user); then the POU will be updated in the current project immediately and an appropriate message will pop up.
Before any Compile	Before any compile process in CoDeSys

Check out

The data base function 'Check out' means that the POU will be marked as 'in the works' and will be locked for other users until it will be de-locked again by a 'Check in' or 'Undo check out' command.

If the option Immediately at start of editing is activated, then an object will be checked out automatically as soon as you start to edit it. If the object is currently already checked out by another user (indicated by a red cross before the object name in the CoDeSys object organizer), then a message will pop up.

Check in

The data base function 'Check in' means, that a new version of the object will be created in the data base. The older versions will be kept anyway.

You can activate one or both of the following options to define the time of automatic Checking in:

At Project Save	as soon as the project is saved
After successfull compile	as soon as the project has been compiled without errors

For each of the options 'Get last version', 'Check out' and 'Check in' additionally the option with Query can be activated. In this case, before the corresponding action is carried out, a dialog opens where you still can decide to cancel the action or otherwise confirm it.

The items of the dialog '**Shared objects**' are the same like in the dialog 'Project objects' described above. The settings apply to all objects which are assigned to the data base category 'Shared objects'.

If you do a primary configuration, the configuration dialogs will appear one after the other and you will be guided by a wizard (button Next). So also you will get to the configuration dialog for the Options for Compile Files, see the following chapter. The settings made in the first dialog will automatically be inherited to the other ones. So those just have to be edited if modifications are necessary.

Cancel will close the dialog without saving the done modifications in the currently opened dialog. You return to the main dialog 'Options' 'Project source control'.

If an already existing configuration has been modified, then the new settings (for all three dialogs) can be saved by pressing OK. After that the dialog will be closed and you return to the main dialog 'Options' 'Project source control'.

Options for Compile Files regarding the project data base

This dialog is part of the option settings for the project data base ('Project' 'Options' 'Project source control'). Here you define how the objects of category 'Compile files' will be handled in the data base. (Besides that two further dialogs are available to define this for objects of category 'Project objects' and 'Shared objects', see previous chapter.)

Dialog 'Compile files' in category Project source control

For the input fields TCP/IP-Address, Port, Project name see the description of dialog 'Project objects/Shared objects'.

Create ASCII-symbol information (.sym)

Create binary symbol information (.sdb)

Create boot project

If this option is activated, then whenever a symbol file *.sym (text format) resp. *.sdb (binary format) will be created, this file will be written to the data base automatically. The entries in the symbol file are created like defined in the Project options category 'Symbol configuration'.

If this option is activated, then whenever a boot project will be created, this file will be written to the data base automatically .

If you are doing a primary configuration, the configuration dialogs will appear one after the other, guided by a wizard (button Next). The settings made in the first dialog will automatically be inherited to the other ones. So those just have to be edited if modifications are necessary.

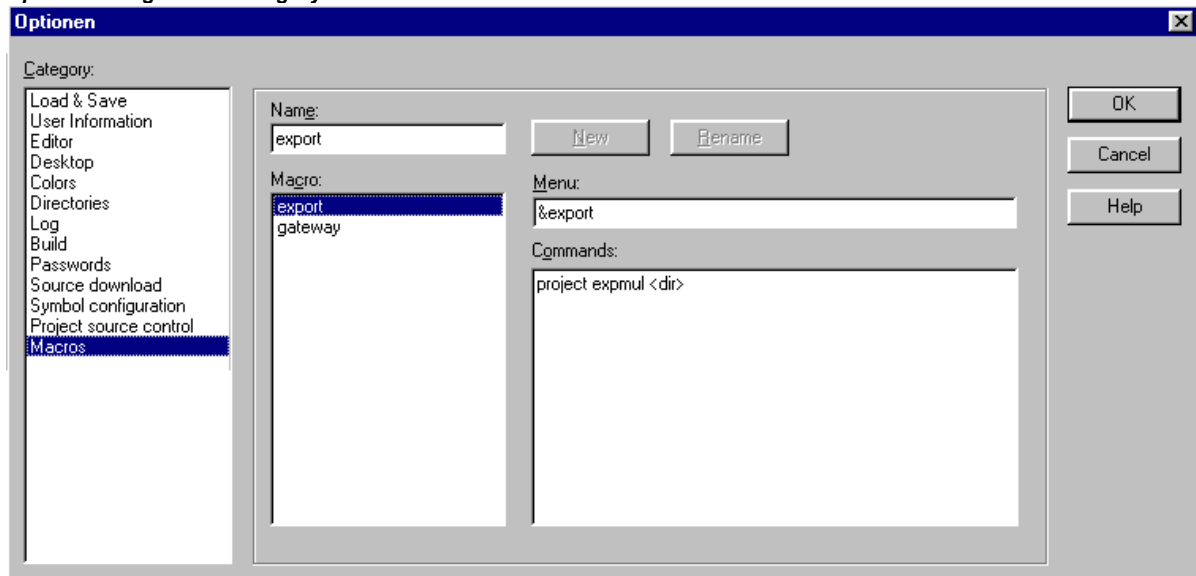
Cancel will close the dialog without saving the done modifications in the currently opened dialog (the settings made in the previous dialogs will be kept anyway). You return to the main dialog 'Options' 'Project source control'.

If you have modified an already existing configuration, then the new settings (for all three dialogs) can be saved by pressing OK. After that the dialog will be closed and you return to the main dialog 'Options' 'Project source control'.

Options for 'Macros'

If you choose this category in the Options dialog, a dialog box opens where you can define macros using the commands of the CoDeSys batch mechanism. These macros afterwards can be called in the 'Edit' 'Macros' menu.

Options dialog for the category Macros



Perform the following steps to define a new macro:

In the input field **Name**, you enter a name for the macro to be created. After the **New** button is pressed, this name is transferred into the **Macrolist** field and marked as selected there. The macro list is represented in a tree structure. The locally defined macros are positioned one below the other. If macro libraries (see below) are integrated, then the library names will be listed and by a mouse-click on the plus- resp. minus-signs in front of those entries you can open or close a list of the library elements.

The **Menu** field is used to define the menu entry with which the macro will appear in the 'Edit' 'Macros' menu. In order to be able to use a single letter as a short-cut, the letter must be preceded by the symbol '&'. Example: the name "Ma&cro 1" generates the menu entry "Macro 1". Example: the name "Ma&cro 1" will create a menu item "Macro 1".

In the editor field **Commands** you define and/or edit the commands that are to constitute the newly created or selected macro. All the commands of the **CoDeSys** batch mechanism and all keywords which are valid for those are allowed. You can obtain a list by pressing the **Help** button. A new command line is started by pressing <Ctrl><Enter>. The context menu with the common text editor functions is obtained by pressing the right mouse button. Command components that belong together can be grouped using quotation marks.

If you want to create further macros, perform steps 1-3 again, before you close the dialog by pressing the OK-button.

If you want to delete a macro, select it in the macro list and press button .

If you want to rename a macro, select it in the macro list, insert a new name in the edit field 'Name' and then press button **Rename**.

To **edit** an existing macro, select it in the macro list and edit the fields 'Menu' and/or 'Commands'. The modifications will be saved when pressing the OK-button.

As soon as the dialog is closed by pressing the **OK**-button the actual description of all macros will be saved in the project.

The macro menu entries in the 'Edit' 'Macros' menu are now displayed in the order in which they were defined. The macros are not checked until a menu selection is made.

Macro libraries:

Macros can be saved in external macro libraries. These libraries can be included in other projects.

Creating a macro library containing the macros of the currently opened project:
 Press button **Create**. You get the dialog **Merge project**, where all available macros are listed. Select the desired entries and confirm with OK. The selection dialog will close and dialog **Save Macrolibrary** will open. Insert here a name and path for the new library and press button **Save**. The library will be created named as **<library name>.mac** and the dialog will be closed.

Including a macro library <library name>.mac in the currently opened project:
 Press button **Include**. The dialog **Open Macrolibrary** will open, which shows files with extension *.mac. Select the desired library and press button **Open**. The dialog will be closed and the library will be added to the tree of the Macrolist.

Hint: The macros of a project also can be exported ('Project' 'Export').

4.3 Managing Projects...

The commands which refer to entire project are found under the menu items 'File' and 'Project'.

'File' 'New'

Symbol: 

With this command you create an empty project with the name "Untitled". This name must be changed when saving.

'File' 'New from template'

Use this command to open any desired CoDeSys project as a "template" project. The dialog for opening a project file will be available and the selected project will be opened with project name "Unknown".

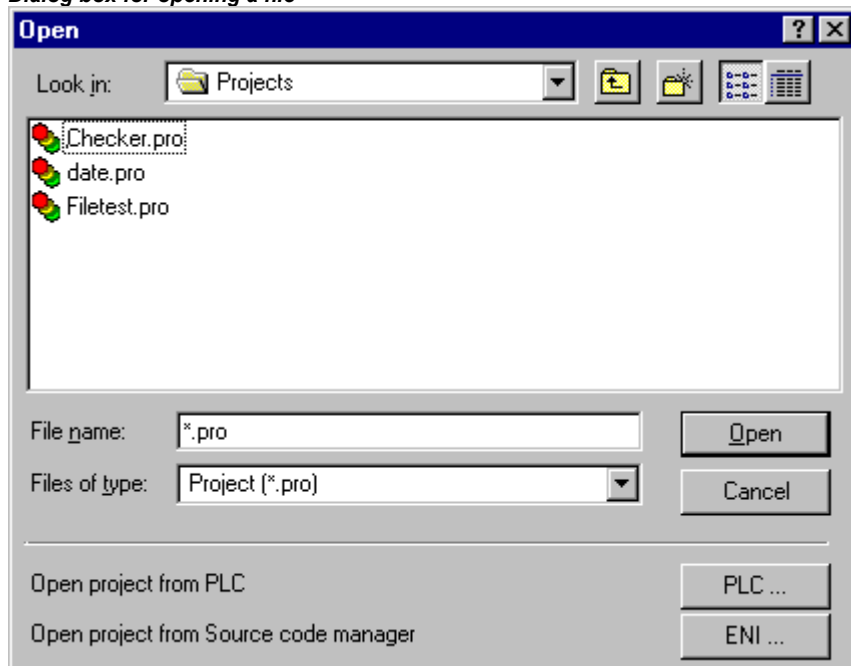
'File' 'Open'

Symbol: 

With this command you open an already existing project. If a project has already been opened and changed, then CoDeSys asks whether this project should be saved or not.

The dialog box for opening a file appears, and a project file with the extension "*.pro" or a library file with the extension "*.lib" must be chosen. This file must already exist. It is not possible to create a project with the command "Open".

Dialog box for opening a file



Open a project from the PLC

To upload a project file from the PLC, press PLC at Open project from PLC. You will obtain, as next, the dialog Communication parameters (see menu 'Online' 'Communication parameters') for setting the transmission parameters when no connection exists yet to the PLC. Once an on-line connection has been created, the system checks whether the same named project files already exist in the directory on your computer hard disc. When this is the case you receive the dialogue Load the project from the controller where you can decide whether the local files should be replaced by those being used by the

controller. (This sequence is the reverse of the sequence of 'Online' 'Load source code', with which the project source file is stored in the controller. Do not confuse with 'Create Boot project'!)

Note: Please note, that you in any case have to give a new name to a project, when you load it from the PLC to your local directory, otherwise it is unnamed. If supported by the target system, a 'Title' entered in the Project info will be pre-defined as new project file name. In this case at loading the project from the PLC the dialog for saving a file will open, where the new file name automatically is entered and can be confirmed or modified.

If there has not yet been loaded a project to the PLC, you get an error message. See also 'Project' 'Options' category 'Sourcedownload'.

Open a project from Source code manager (ENI data base)

To open a project which is stored in a ENI project data base, activate option **Open project from Source code manager** can be used . It is a precondition that you have access to an ENI Server which serves the data base. Press button ENI..., to get a dialog where you can connect to the server concerning the data base category 'Project objects'.

Insert the appropriate access data (TCP/IP-Address, Port, User name, Password, Read only) and the data base folder (Project name) from which the objects should be get and confirm with **Next**. The dialog will be closed and another one will open where you have to insert the access data for the data base category 'Shared objects'. If you press button **Finish** the dialog will be closed and the objects of the defined folders will automatically be retrieved and displayed in the CoDeSys Object manager. If you want to continue to keep the project objects under data base control, then open the Project options dialogs to set the desired parameters.

The access data are stored in the codesys.ini file, user name and password however only if the project option 'Save ENI credentials' (see Chapter 4.2, Category Load & Save) is activated.

Most recently opened files

The most recently opened files are listed in the Files menu below the command 'File' 'Exit'. If you choose one of them, then this project is opened.

If Passwords or User groups have been defined for the project, then a dialog box appears for entering the password.

'File' 'Close'

With this command you close the currently-open project. If the project has been changed, then CoDeSys asks if these changes are to be saved or not.

If the project to be saved carries the name "Untitled", then a name must be given to it (see 'File' 'Save as').

'File' 'Save'

Symbol:  **Shortcut: <Ctrl>+<S>**

With this command you save any changes in the project. If the project to be saved is called "Untitled", then you must give it a name (see 'File' 'Save as').

'File' 'Save as'

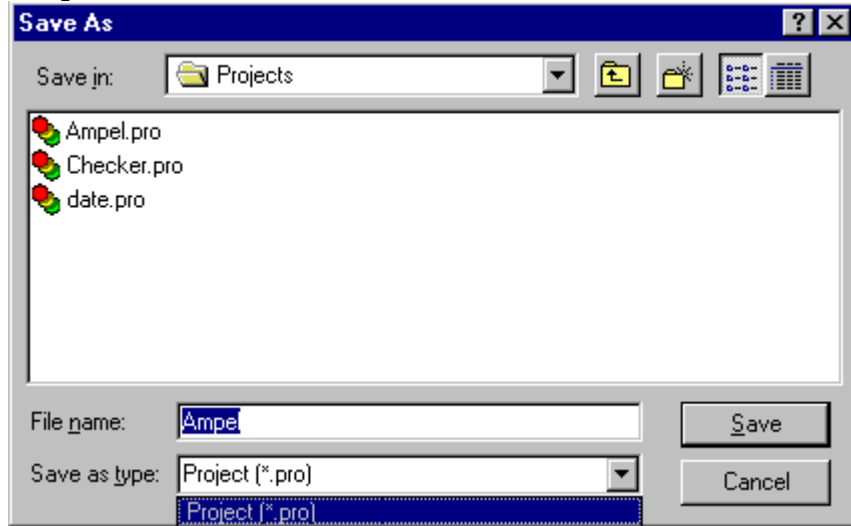
With this command the current project can be saved in another file or as a library. This does not change the original project file.

After the command has been chosen the Save dialog box appears. Choose either an existing **File name** or enter a new file name and choose the desired **file type**.

If the project is to be saved under a new name, then choose the file type **CoDeSys Project (*.pro)**.

If you choose the file type **Project Version 1.5 (*.pro), 2.0 (*.pro), 2.1 (*.pro) or 2.2 (*.pro)**, then the current project is saved as if it were created with the version 1.5, 2.0, 2.1 or 2.2. Specific data of the version 2.3 can thereby be lost! However, the project can be executed with the version 1.5, 2.0, 2.1 or 2.2.

Dialog box for Save as

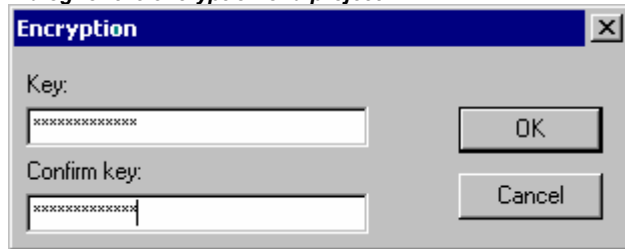


You can also save the current project as a library in order to use it in other projects. Choose the file type **Internal library (*.lib)** if you have programmed your POU's in CoDeSys.

Choose the file type **External library (*.lib)** if you want to implement and integrate POU's in other languages (e.g. C). This means that another file is also saved which receives the file name of the library, but with the extension ".h". This file is constructed as a C header file with the declarations of all POU's, data types, and global variables. If external libraries are used, in the simulation mode the implementation, written for the POU's in CoDeSys, will be executed. Working with the real hardware the implementation written in C will be executed.

In order to save the project as an encrypted project or library, choose option **Encrypted CoDeSys Projekt (*.pro)** resp. **Encrypted internal library (*.lib)** or **Encrypted external library (*.lib)**. In this case you get the 'Encryption' dialog, where you can define and confirm a key. The project later cannot be opened esp. a library cannot be used without this key :

Dialog for the encryption of a project

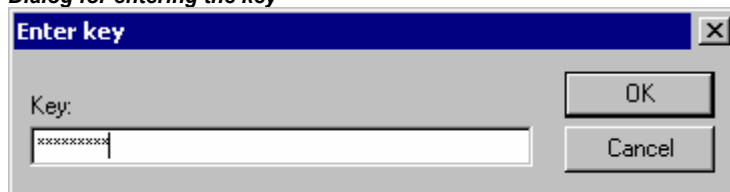


The encryption extends the protection of a project, which up to now was only possible via the assignment of passwords for access and write protection. These possibilities will exist further on, but regard that they e.g. cannot avoid that a library is included in a project without the need of entering a library password (key).

A key once defined will be saved with any further savings of the project. To modify that key, you again have to use the 'Save as' dialog.

If an encrypted project should be opened resp. if an encrypted library should be used in a project, the dialog asking for the key will appear.

Dialog for entering the key



Licensing a library:

If you want save the project as a licensed library, you can add the appropriate licensing information in the dialog 'Edit Licensing Information'. Open the dialog by pressing the button **Edit license info...** See for a description in 'License Management in CoDeSys'.

After having done all settings, press **OK**. The current project will be saved in the indicated file. If the new file name already exists, then you are asked if you want to overwrite this file.

When saving as a library, the entire project is compiled. If an error occurs thereby, then you are told that a correct project is necessary in order to create a library. The project is then not saved as a library.

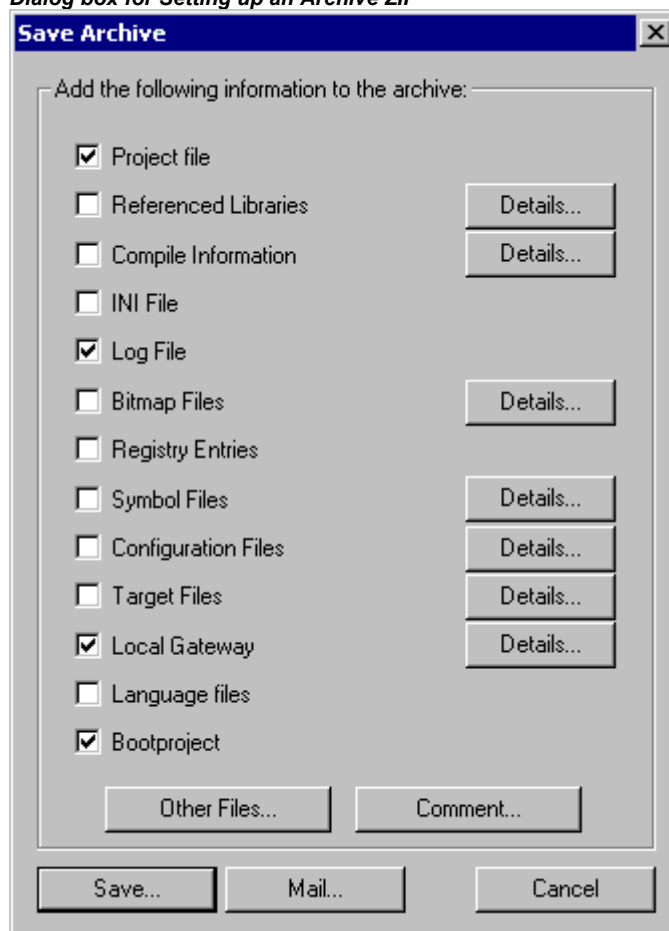
'File' 'Save/Mail Archive'

This command is used to set up and create a project archive file. All files which are referenced by and used with a CoDeSys project can be packed in a compressed zip file. The zip file can be stored or directly can be sent in an email. This is useful if you want to give forward a set of all project relevant files.



Please regard: The archive function is not practical for restoring a project environment. It is designated for an easy packing of all files belonging to a project. When unpacking an archive the paths of the particular files must be adapted to the actual CoDeSys environment !

When the command is executed, the dialog box Save Archive opens:

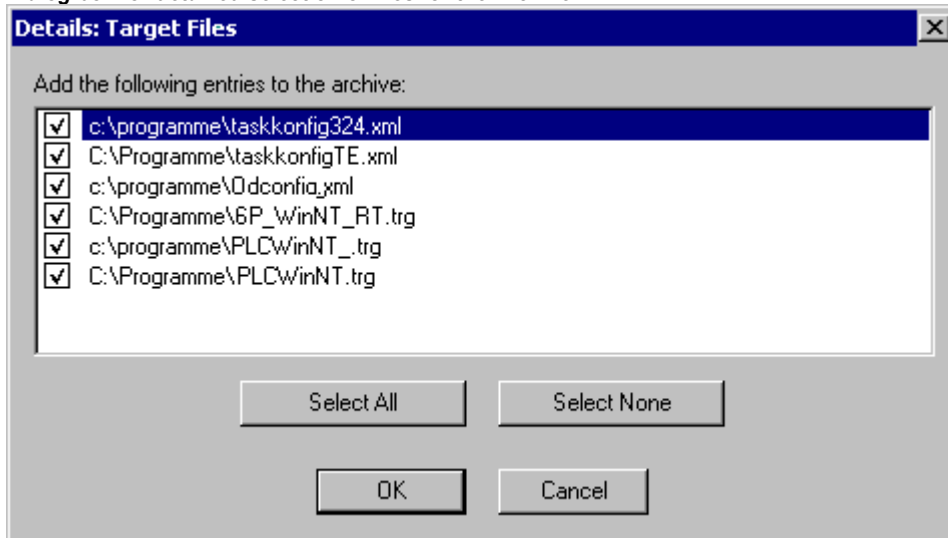
Dialog box for Setting up an Archive ZIP



Here you can define which file categories should be added to the archive zip file: Select or deselect a category by activating/deactivating the corresponding checkbox. Do this by a single mouse click in the


checkbox or by a double-click on the category name. If a category is marked with , all files of this category will be added to the zip file, if it is marked with , none of the files will be added. To select single files of a category press the corresponding button **Details**. The dialog Details will open with a list of available files.

Dialog box for detailed selection of files for the Archive ZIP



The dialog shows a list of all files which are available for the category: Automatically alle files are selected, an exception is category 'Target Files' where only the files are selected which are relevant for the currently set target system. For modifying the selection activate resp. deactivate the desired files. With the button **Select All** all files of the list are selected, with **Select None** none of them. A single file can be selected/deselected by a mouse click in the checkbox, also by a double-click on the list entry or by pressing the spacebar when the list entry is marked.

Close the Details dialog with **Save** to store the new settings.

In the main dialog the checkbox of categories, for which not all files are selected, will appear with a grey background color . The following file categories are available, the right column of the table shows which files can be added to the zip file:

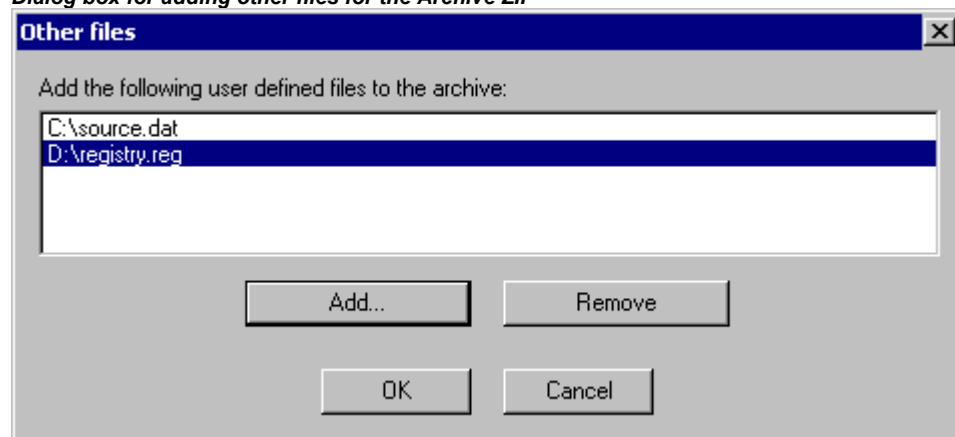
Kategorie	Dateien
Project File	projectname.pro (the CoDeSys project file)
Referenced Libraries	*.lib, *.obj, *.hex (libraries and if available the corresponding object and hex-files)
Compile Information	*.ci (compile information), *.ri (download/reference information) <temp>.* (temporary compile and download files) also for simulation
INI File	Codesys.ini
Log File	*.log (project log file)
Bitmap Files	*.bmp (bitmaps for project POU's and visualizations)
Registry Entries	Registry.reg (Entries for Automation Alliance, Gateway und SPS; the following subtrees will be packed: HKEY_LOCAL_MACHINE\SOFTWARE\3S-Smart Software Solutions HKEY_LOCAL_MACHINE\SOFTWARE\AutomationAlliance")
Symbol Files	*.sdb, *.sym (symbolic information)

Configuration files	files used for PLC configuration (configuration files, device files, icons etc.): e.g. *.cfg, *.con, *.eds, *.dib, *.ico
Target Files	*.trg (target files in binary format for all installed targets) *.txt (target files for the installed targets in text format, if available)
Local Gateway	Gateway.exe, GatewayDDE.exe, GClient.dll, GDrvBase.dll, GDrvStd.dll, GHandle.dll, GSymbol.dll, GUtil.dll, further DLLs in the gateway directory if available
Language Files	language files used for visualizations (*.vis, *.xml)
Bootproject	Bootproject files <project name>.prg, <project name>.chk resp. the target specific bootproject files.

To add any other files to the zip, press the button **Other Files**. The dialog 'Other files' will open where you can set up a list of desired files.

Press the button Add to open the standard dialog for opening a file, where you can browse for a file. Choose one and confirm with Open. The file will be added to the list in the 'Other files' dialog. Repeat this for each file you want to add. To delete entries from the list, press the button Remove. When the list of selected files is ok, close the dialog with OK.

Dialog box for adding other files for the Archive ZIP



To add a Readme file to the archive zip, press the button **Comment**. A text editor will open, where you can enter any text. If you close the dialog with OK, during creation of the zip file a **readme.txt** file will be added. Additionally to the entered comments it will contain information about the build date and version of CoDeSys.

If all desired selections have been made, in the main dialog press

Save... to create and save the archive zip file: The standard dialog for saving a file will open and you can enter the path, where the zip should be stored. The zip file per default is named **<projectname>.zip**. Confirm with **Save** to start building it. During creation the current progress status is displayed and the subsequent steps are listed in the message window. There also a message will be displayed if any file could not not been found.

Mail... to create a temporary archive zip and to automatically generate an empty email which contains the zip as an attachment. This feature only works if the MAPI (Messaging Application Programming Interface) has been installed correctly on the system, otherwise an error message is generated. During setup of the email the progressing status is displayed and the steps of the action are listed in the message window. The temporary zip file will be removed automatically after the action has been finished.

Cancel to cancel the action; no zip file will be generated.

'File' 'Print'

Shortcut: <Ctrl>+<P>

With this command the content of the active window is printed.

After the command has been chosen, then the Print dialog box appears. Choose the desired option or configure the printer and then click **OK**. The active window is printed. Color output is available from all editors.

You can determine the **number of the copies** and print the version to a file.

With the button **Properties** you open the dialog box to set up the printer.

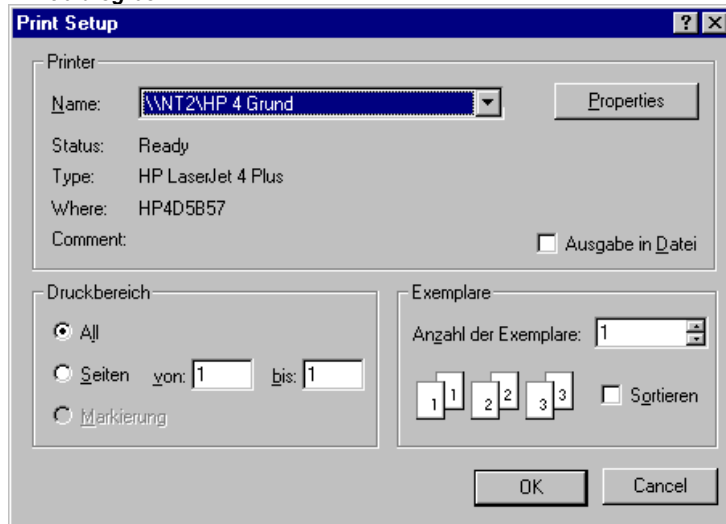
You can determine the layout of your printout with the command 'File' 'Printer Setup'.

During printing the dialog box shows you the number of pages already printed. When you close this dialog box, then the printing stops after the next page.

In order to document your entire project, use the command 'Project' 'Document'.

If you want to create a document frame for your project, in which you can store comments regarding all the variables used in the project, then open a global variables list and use the command '**Extras**' '**Make docuframe file**'.

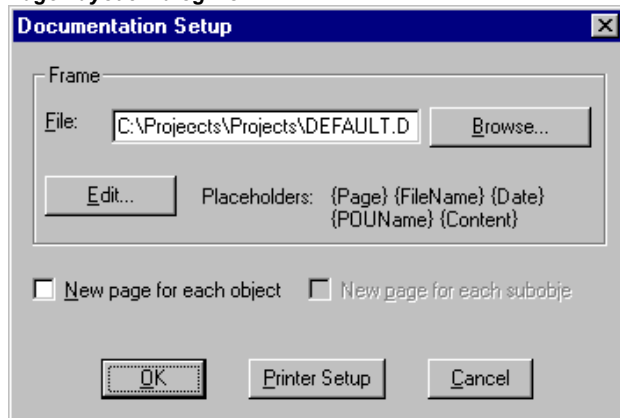
Print dialog box



'File' 'Printer setup'

With this command you can determine the layout of the printed pages. The following dialog box is now opened:

Page Layout Dialog Box



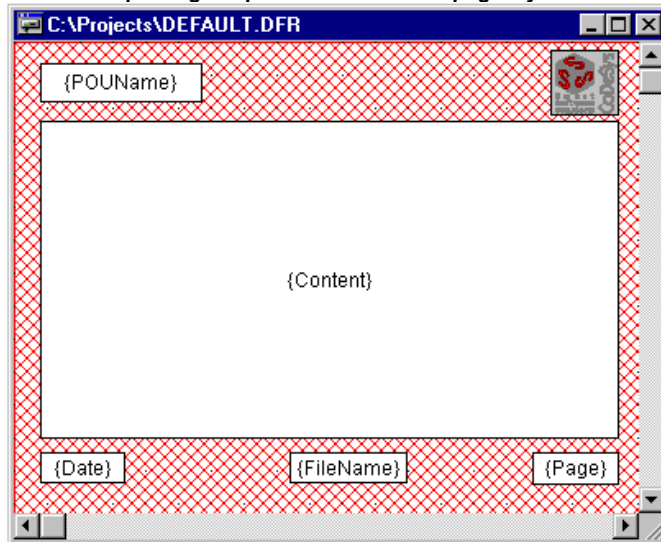
In the field **File** you can enter the name of the file with the extension ".dfr" in which the page layout should be saved. The default destination for the settings is the file DEFAULT.DFR.

If you would like to change an existing layout, then browse through the directory tree to find the desired file with the button **Browse**

You can also choose whether to begin a **new page for each object** and **for each subobject**. Use the **Printer Setup** button to open the printer configuration.

If you click on the **Edit** button, then the frame for setting up the page layout appears. Here you can determine the page numbers, date, filename and POU name, and also place graphics on the page and the text area in which the documentation should be printed.

Window for pasting the placeholders on the page layout



With the menu item **'Insert' 'Placeholder'** and subsequent selection among the five placeholders (**Page, POU name, File name, Date, and Content**), insert into the layout a so-called placeholder by dragging a rectangle on the layout while pressing the left mouse button. In the printout they are replaced as follows:

Command	Placeholder	Effect
Page	{Page}	Here the current page number appears in the printout.
POU name	{POU Name}	Here the current name of the POU appears.
File name	{File Name}	Here the name of the project appears.
Date	{Date}	Here the current date appears.
Contents	{Contents}	Here the contents of the POU appear.

In addition, with **'Insert' 'Bitmap'** you can insert a bitmap graphic (e.g. a company logo) in the page. After selecting the graphic, a rectangle should also be drawn here on the layout using the mouse. Other visualization elements can be inserted (see Visualizations).

If the template was changed, then CoDeSys asks when the window is closed if these changes should be saved or not.

Hint: In order to be aware of the page format which will be valid for printouts, define the layout as described above and additionally activate option 'Show print area margins' in the project options, category Desktop.

'File' 'Exit'**Shortcut: <Alt>+<F4>**

With this command you exit from CoDeSys.

If a project is opened, then it is closed as described in 'File' 'Save'.

'Project' 'Build'**Shortcut: <F11>**

The project is compiled using 'Project' 'Build'. The compilation process is basically incremental, that is only changed POU's are recompiled. The necessary information about the last compilation is stored in a *.ci-file when the project is saved. A non-incremental compilation can also be obtained if the command 'Project' 'Clear all' is first executed.

For target systems that support Online Change, all POU's that will be loaded into the controller on the next download are marked with a blue arrow in the Object Organizer after compilation.

The compilation process that is carried out with 'Project' 'Build' occurs automatically if the controller is logged-in via 'Online' 'Log-in'.

See a diagram showing the relations between Project-Build, Project-Download, Online Change and Login on the target system in Chapter 4.6, 'Online', 'Login'.

During compilation a message window is opened which shows the progress of the compilation process, any errors and warnings which may occur during compilation as well as information on the used POU indices and memory space (number and percentage). Errors and warnings are marked with numbers. Using F1 you get more information about the currently selected error.

See the listing of all available error messages and warnings.

Example for error messages and compile information in the message window of a project

```
Interface of POU 'PLC_PRG_TRD'
Error 4024: PLC_PRG_TRD (5): Expecting ';' or '=' before 'bAlarm1'
Error 3781: PLC_PRG_TRD (5): 'END_VAR' or identifier expected
2 Error(s), 0 Warning(s).
```

```
Declarations of the global constants
Declarations of the global library constants
Interface of POU 'PLC_PRG'
Interface of POU 'PLC_PRG_TRD'
Declarations of the global variables
Data allocation
Check task configuration
Implementation of POU 'PLC_PRG'
Implementation of POU 'PLC_PRG_TRD'
Implementation of task 'PLC_PRG_TASK'
Check of the parameter configuration
Hardware-Configuration
POU indices:19 (3%)
Size of used data: 636 of 2097152 bytes (0.03%)
Size of used retain data: 0 of 32768 bytes (0.00%)
0 Error(s), 0 Warning(s).
```

If the option **Save before compilation** is selected in the options dialog of the Load & Save category, the project is stored before compilation.

An object selected in the Object Organizer resp. several objects can be excluded from compilation by command 'Exclude from build' which is available in the context menu, resp. via an appropriate configuration ('Exclude objects') in the Options for Build (see chapter 4.2, Options for Build).

Note: Cross references are created during compilation and are stored with the compilation information. In order to be able to use the command 'Show Call Tree', resp. to get up to date results with the commands 'Show Cross Reference', 'Unused Variables', 'Overlapping memory areas', 'Concurrent Access', and 'Multiple Write Access on output' ('Project' 'Check' menu), the project must be rebuilt after any change.

'Project' 'Rebuild all'

With 'Project' 'Rebuild all', unlike the incremental compilation ('Project' 'Build'.Project..Build.>Proc), the project is completely recompiled. Download information is not discarded, however, as is the case with the command 'Clear All'. Regard the possibility to exclude objects from compilation (see chapter 4.2, Options for Build).

See a diagram showing the relations between Project-Build, Project-Download, Online Change and Login on the target system in Chapter 4.6, 'Online' 'Login'.

'Project' 'Clean all'

With this command, all the information from the last download and from the last compilation is deleted.

After the command is selected a dialog box appears, reporting that Login without new download is no longer possible. At this point the command can either be cancelled or confirmed.

Note: After having done a 'Clean all', a login on the PLC project is only possible if the *.ri file with the project information from the last download was first renamed or saved outside the project directory (see 'Load download information') and can now be reloaded explicitly prior to logging-in.

See a diagram showing the relations between Project-Build, Project-Download, Online Change and Login on the target system in Chapter 4.6, 'Online' 'Login'.

'Project' 'Load download information'

With this command the download information belonging to the project can get reloaded. After choosing the command the standard dialogue 'File Open' opens.

The download information is saved automatically at each download and, dependent on the target system, potentially also at each offline creation of a boot project to a file, which is named **<project name><target identifier>.ri** and which is put to the project directory. This file gets reloaded each time the project is reopened and at login it is used to check the code of which POU's has been changed. Only these POU's will then be loaded to the PLC during online change procedure. Thus the *.ri-file is a precondition for an Online Change.

But: If the *.ri-file in the project directory gets deleted by the command '**Project' 'Clean all'**', you only can reload the download information if you had stored the *.ri-file in another directory too. See hints in Chapter 4.6, 'Online' 'Login'.

See also in Chapter 4.6, 'Online' 'Login' a diagram showing the relations between Project-Build, Project-Download, Online Change and Login on the target system.

'Project' 'Translate into another language'

This menu item is used for translating the current project file into another language. This is carried out by reading in a translation file that was generated from the project and externally enhanced in the desired national language with the help of a text editor. The project can be just displayed or really get translated into one of the generated language versions.

The following menu sub-items are present:

Create translation file

Translate project

Show project translated

Toggle translation

See also: 'Editing of the translation file'

Create translation file

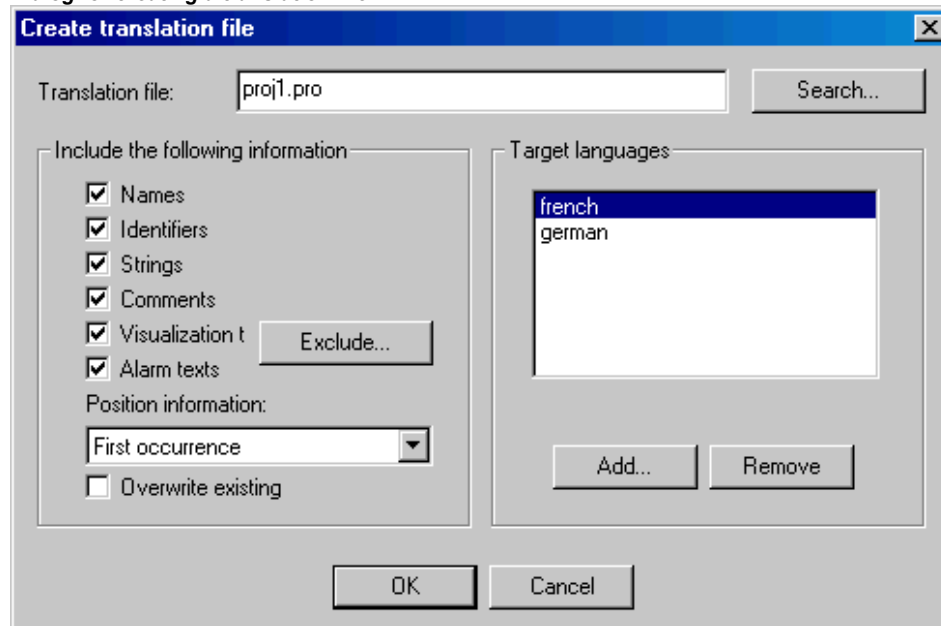
This command in the 'Project' 'Translate into another language' menu leads to the 'Create translation file' dialog.

In the **Translation file** field, enter a path that shows where the file is to be stored. The default file extension is *.tlt; this is a text file. You also can use the extension *.txt, which is recommended, if you want to work on the file in EXCEL or WORD, because in this case the data are organized in table format.

If there already exists a translation file which you want to process, give the path of this file or use the **Search** button to reach the standard Windows file selection dialog.

The following information from the project can optionally be passed to the translation file that is being modified or created, so that they will be available for translation: **Names** (names, e.g. the title 'POUs' in Object Organizer), **Identifiers**, **Strings**, **Comments**, **Visualization texts**, **Alarm texts**. In addition, **Position information** for these project elements can be transferred.

Dialog for creating a translation file



If the corresponding options are checked, the information from the current project will be exported as language symbols into a newly created translation file or added to an already existing one. If the respective option is not selected, information belonging to the pertinent category, regardless of which project it came from, will be deleted from the translation file.

Texts in Visualizations:

The option "Visualisation texts" only concerns 'Text' and 'Tooltip-Text' of a visualization element. Regard the following items when using a translation file for visualization texts:

A *.tlt- or *.txt translation file only can be used with CoDeSys or CoDeSys HMI, not however with Target-Visualization or Web-Visualization. It might be better to use a special visualization language file *.vis.

Switching to another language is only possible in online mode. This means that the visualization texts will not be translated by command 'Translate into another language'. A language change can only occur in Online mode if the corresponding language is entered in the 'Extras' 'Settings' dialog.

If a *.tlt- or *.txt-file should be used for visualization texts ('Text' and 'Text for Tooltip'), the texts must be bracketed by two "#" symbols in the configuration dialog of the visualization element (e.g. #text#) in order to be transferred to the translation file.

Concerning the language switching in visualizations see the user manual for the CoDeSys Visualization.

Position information: This describes with the specifications file path, POU and line the position of the language symbol made available for translation. Three options are available for selection:

'None': No position information is generated.

'First appearance': The position on which the element first appears is added to the translation file.

'All': All positions on which the corresponding element appears are specified.

If a translation file created earlier is to be edited which already contains more position information than that currently selected, it will be correspondingly truncated or deleted, regardless of which project it was generated from.

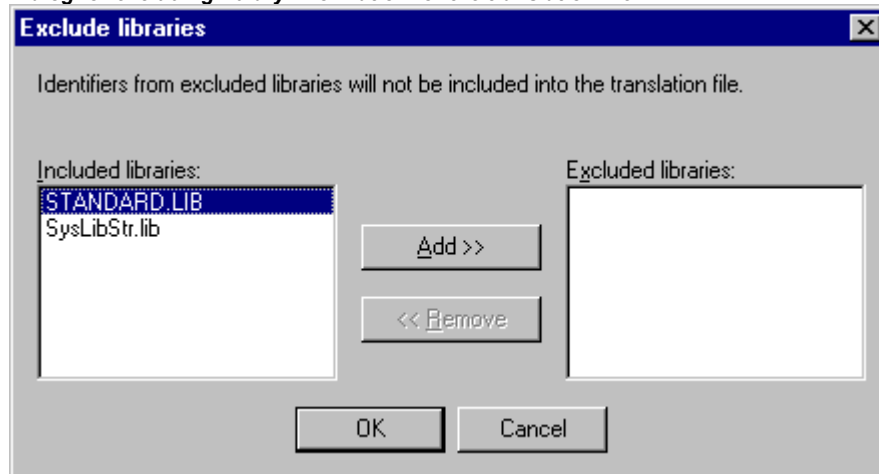
Note: A maximum of 64 position specifications will be generated per element (language symbol), even if the user has selected "All" under "Position Information" in the 'Create Translation File' dialog.

Overwrite existing: Existing position information in the translation file, that is currently being processed, will be overwritten, regardless of which project generated it.

Target languages: This list contains identifiers for all languages which are contained in the translation file, as well as those to be added upon completion of the 'Create translation file' dialog.

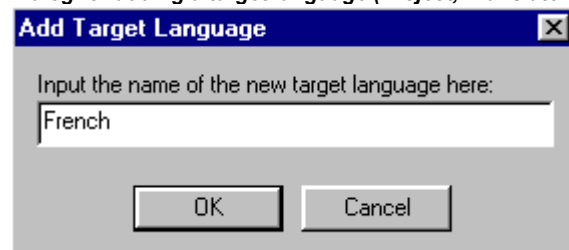
The **Exclude** button opens the 'Exclude libraries' dialog. Here, libraries included to the project can be selected, whose identifier information is not to be transferred to the translation file. To accomplish this, the corresponding entry in the table **Included libraries** on the left is selected with the mouse and placed in the **Excluded libraries** table to the right using the **Add** button. Likewise, entries already placed there can be removed using the **Remove** button. OK confirms the setting and closes the dialog.

Dialog for excluding library information for the translation file



The **Add** button opens the 'Add Target Language' dialog:

Dialog for adding a target language (Project, Translate into Another Language)



A language identifier must be entered into the editor field; it may not have a space or an umlaut character (ä, ö, ü) at either the beginning or the end.

OK closes the 'Add Target Language' dialog and the new target language appears in the target language list.

The **Remove** button removes a selected entry from the list.

You may also confirm the "Create translation file" dialog via **OK**, in order to generate a translation file.

If a translation file of the same name already exists you will get the following confirmation message to be answered Yes or No:

```
" The specified translation file already exists. It will now be altered and a
  backup copy of the existing file will be created. Do you want to continue?"
```

No returns you without action to the 'Create translation file' dialog. If **Yes** is selected, a copy of the existing translation file with the filename "Backup_of_<translation file>.xlt" will be created in the same directory and the corresponding translation file will be modified in accordance with the options that have been entered.

The following takes place when a translation file is generated:

For each new target language, a placeholder ("##TODO") is generated for each language symbol to be displayed. (See 'Editing of the translation file' for how to work on the translation file.)

If an existing translation file is processed, file entries of languages that appear in the translation file, but not in the target language list, are deleted, regardless of the project from which they were generated.

Editing of the translation file

The translation file must be opened and saved as a text file. The signs ## mark keywords. The ##TODO-placeholders in the file can be replaced by the valid translation. For each language symbol a paragraph is generated which starts and ends with a type identifier. For example ##NAME_ITEM and ##END_NAME_ITEM include a section for the name of an object as used in the object organizer. (COMMENT_ITEM marks sections for comments, IDENTIFIER_ITEM those for identifiers, STRING_ITEM those for strings and VISUALTEXT_ITEM those for visualization texts).

See in the following an example of a translation file paragraph which handles the name of one of the POU's of the project. ST_Visu. The target languages shall be English(USA) and French. In this example the position information of the project element which should be translated has been added:

- before translation:

```
##NAME_ITEM
[D:\CoDeSys\projects\Bspdt_22.pro::ST_Visualisierung::0]
ST_Visualisierung
##English :: ##TODO
##French :: ##TODO
##END_NAME_ITEM
```

- after translation:

The ##TODOs have been replaced by the English resp. French word for 'Visualisierung':

```
##NAME_ITEM
[D:\CoDeSys\projects\Bspdt_22.pro::ST_Visualisierung::0]
ST_Visualisierung
##English :: ST_Visualization
##French :: ST_Visu
##END_NAME_ITEM
```

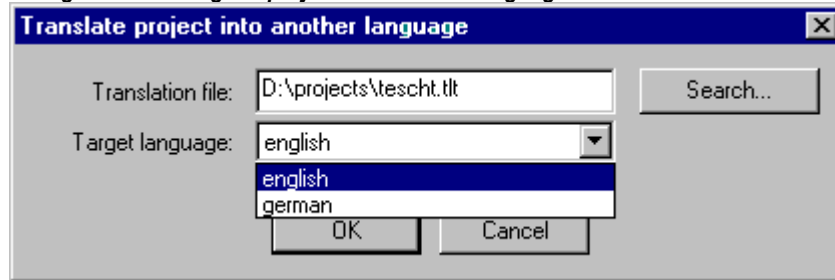
Please check that the translated Identifier and Names remain valid concerning the standard and that strings and comments are in correct brackets. Example: For a comment (##COMMENT_ITEM) which is represented with "(* Kommentar 1)" in the translation file, the "##TODO" behind "##English" must be replaced by a "(* comment 1 *)". For a string (##STRING_ITEM) represented with "zeichenfolge1" the "##TODO" must be replaced by "string1".

Hint: The following parts of a translation file should not be modified without detailed knowledge: Language block, Flag block, Position information, Original texts.

Translate Project (into another Language)

This command in the 'Project' 'Translate into Another Language' menu opens the 'Translate Project into Another Language' dialog.

Dialog for translating the project into another language



The current project can be translated into another language if an appropriate translation file is used.

Note: If you want to save the version of the project in the language in which it was originally created, save a copy of the project prior to translation under a different name. **The translation process cannot be undone.** Consider in this context the possibility just to display the project in another language (in this display version then however not editable).

In the field **Translation file**, provide the path to the translation file to be used. By pressing **Search** you may access the standard Windows file selection dialog.

The field **Target language** contains a list of the language identifiers entered in the translation file, from which you can select the desired target language.

OK starts the translation of the current project into the chosen target language with the help of the specified translation file. During translation, a progress dialog is displayed, as well as error messages, if any. After translation, the dialog box and all open editor windows of the project are closed.

Cancel closes the dialog box without modification to the current project.

If the translation file contains erroneous entries, an error message is displayed after OK is pressed, giving the file path and the erroneous line, e.g.: "[C:\Programs\CoDeSys\projects\visu.tlt (78)]; Translation text expected"

Note: Please regard the specialities for texts in visualizations; see above, 'Create translation file'.

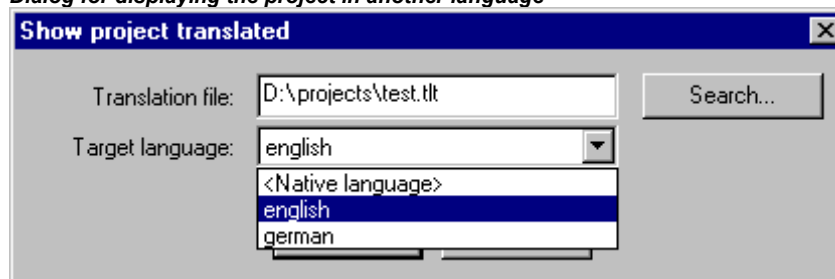
'Show project translated'

If there is a translation file available for the project, you can display one of the language versions defined there, without overwriting the original language version of the project.

(Regard this possibility in comparison to the "real" translating of a project, which you would do with the command 'Translate Project', and which would mean to create a new version of the project !)

The command 'Translate this project' in menu 'Project' 'Translate into another language' opens the dialog 'Translate project into another language'.

Dialog for displaying the project in another language



In field **Translation file** insert the path of the translation file, you want to use. You can receive assistance by the standard dialog for opening a file which is opened by button **Browse**.

In field **Target language** you find a selection list, which besides the entry "<Native language>" also offers the language identifiers which are defined by the currently set translation file. The original language is that one, which is currently saved with the project. (It only could be changed by a 'Project' 'Translate'.) Choose one of the available languages and confirm the dialog with OK. Thereupon the project will be displayed in the chosen language, **but cannot be edited in this view !**

If you want to change back to viewing the project in its original language, use command 'Switch translation'.

Note: Please regard the specialities for texts in visualizations; see above, 'Create translation file'.

'Toggle translation'

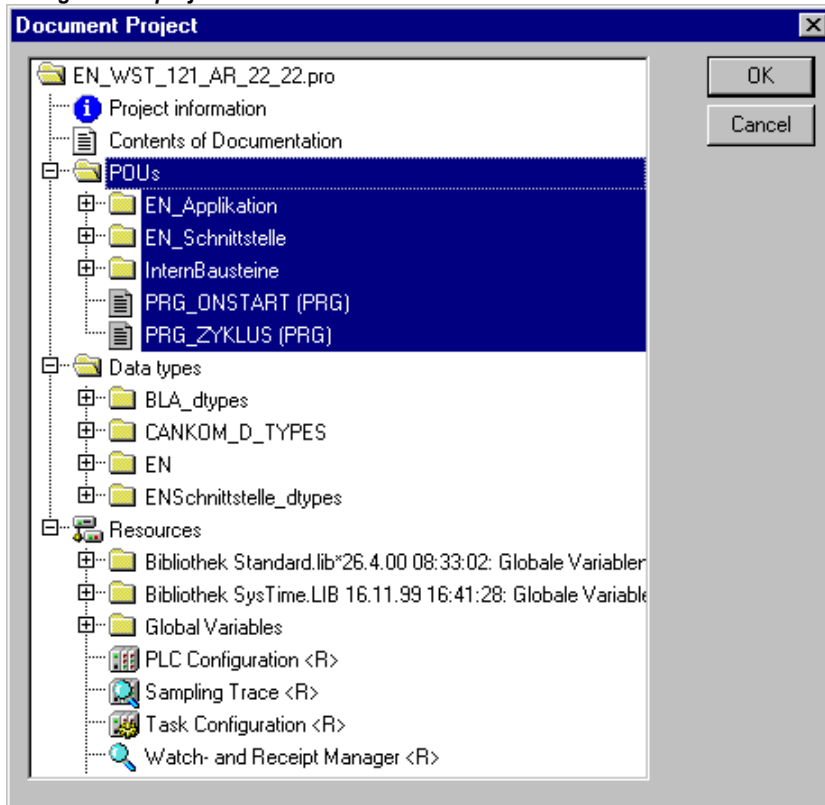
If you have got displayed the project (not editable) in another language by command 'Show project translated', you can now switch between this language version and the (editable) original version by using the command 'Switch translation' of menu 'Project' 'Translate (into another Language)' .

Note: Please regard the specialities for texts in visualizations; see above, 'Create translation file'.

'Project' 'Document'

This command lets you print the documentation of your entire project.

Dialog box for project documentation



The elements of a complete documentation are:

The POU's,

the contents of the documentation,

the data types,

the visualizations,

the resources (Global Variables, Variables Configuration, Sampling Trace, PLC Configuration, Task Configuration, Watch- and Receipt manager)

the call trees of POU's and data types, as well as

the cross reference list.

For the last two items the project must have been built without errors. Only those areas in the dialog box are printed which are highlighted in blue.

If you want to select the entire project, then select the name of your project in the first line.

If, on the other hand, you only want to select a single object, then click on the corresponding object or move the dotted rectangle onto the desired object with the arrow key. Objects which have a plus sign in front of their symbols are organization objects which contain other objects. With a click on a plus sign organization object is expanded, and with a click on the resulting minus sign it can be closed up again. When you select an organization object, then all relevant objects are also selected. By pressing the <Shift> key you can select a group of objects, and by pressing the <Ctrl> key you can select several individual objects.

Once you have made your selection, then click on **OK**. The Print dialog box appears. You can determine the layout of the pages to be printed with 'File' 'Printer setup'.

'Project' 'Export'

With **CoDeSys** projects can be exported or imported. That allows you to exchange programs between different IEC programming systems.

There is a standardized exchange format for POU's in IL, ST, and SFC (the Common Elements format of IEC 1131-3). For the POU's in LD and FBD and the other objects **CoDeSys** has its own filing format since there is no text format for this in IEC 1131 3.

The selected objects are written to an ASCII file.

POU's, data types, visualizations, and the resources can be exported. In addition, entries in the library manager, that is the linking information to the libraries, can be exported (not the libraries themselves!).

Important: Re-importing an exported FBD or LD POU results in an error if a comment in the graphical editor contains a single quotation mark ('), as this will be interpreted as the beginning of a string !

Once you have made your selection in the dialog box window (the same way as with 'Project' 'Document'), you can decide, whether you want to export the selected parts to one file or to export in separate files, one for each object. Switch on or off the option **One file for each object** then click on **OK**. The dialog box for saving files appears. Enter a file name with the expansion ".exp" respectively a directory for the object export files, which then will be saved there with the file name <objectname.exp>.

'Project' 'Import'

In the resulting dialog box for opening files select the desired export file.

The data is imported into the current project. If an object with the same name already exists in the same project, then a dialog box appears with the question "Do you want to replace it?": If you answer **Yes**, then the object in the project is replaced by the object from the import file. If you answer **No**, then the name of the new objects receives as a supplement an underline and a digit ("_0", "_1", ..). With **Yes, all** or **No, all** this is carried out for all objects.

If the information is imported to link with a library, the library will be loaded and appended to the end of the list in the library manager. If the library was already loaded into the project, it will not be reloaded. If, however, the export file that is being imported, shows a different storage time for the library, the library name is marked with a "*" in the library manager (e.g. standard.lib*30.3.99 11:30:14), similar to the loading of a project. If the library can not be found, then an information dialog appears: "Cannot find library {<path>}<name> <date> <time>", as when a project is loaded.

In the message window the import is registered.

'Project' 'Siemens Import'

In the submenu "Siemens Import" you find the commands for importing POU's and variables from Siemens-STEP5 files.

The following commands are available:

"Import from SEQ symbol file"

"Import from S5 file"

See Appendix G: for more detailed information about Siemens import.

'Project' 'Compare'

This command is used to compare two projects or to compare the actual version of one project with that which was saved last.

Overview:

- Definitions:
- actual project:** Project, which you are currently working on.
 - reference project:** Project, which should be compared with the actual project.
 - compare mode:** in this mode the project will be displayed after the command 'Project' 'Compare' has been executed.
 - unit:** Smallest unit which can be compared. Can be a line (declaration editor, ST editor, IL editor), a network (FBD editor, LD editor) or a element/POU (CFC editor, SFC editor).

In compare mode the actual project and the reference project will be presented in a bipartite window. The names of the POU's, for which differences have been found, are marked by color. For editor POU's also the content of the POU's is displayed in a vis-a-vis way. The results and the way of presenting in compare mode depend on: 1. what filters have been activated for the compare run, affecting the consideration of white spaces and comments during comparison; 2. whether modification within lines or networks or elements are evaluated as a completely new inserting of a POU or not.

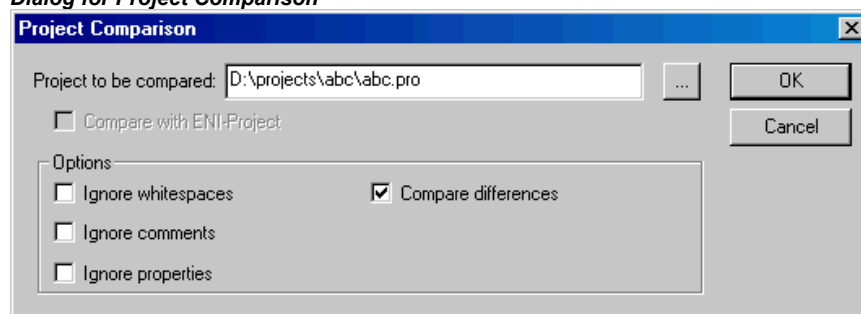
The version of the reference project can be accepted for single differences or for 'all equally marked' differences. To accept means that the version of the reference project is taken over to the actual project.


Please note: In compare mode (see status bar: COMPARE) the project cannot get edited !

Execute comparison

After executing the command 'Project' 'Compare' the dialog **Project Comparison** opens:

Dialog for Project Comparison



Insert the path of the reference project at **Project to compare**. Press button  if you want to use the standard dialog for opening a project. If you insert the name of the actual project, the current version of the project will be compared with the version which was saved last.

If the project is under source control in an ENI data base, then the local version can be compared with the actual version found in the data base. For this activate option **Compare with ENI-Project**.

The following options concerning the comparison can be activated:

Ignore whitespaces: There will be detected no differences which consist in a different number of whitespaces.

Ignore comments: There will be detected no differences in comments.

Ignore properties: There will be detected no differences in object properties.

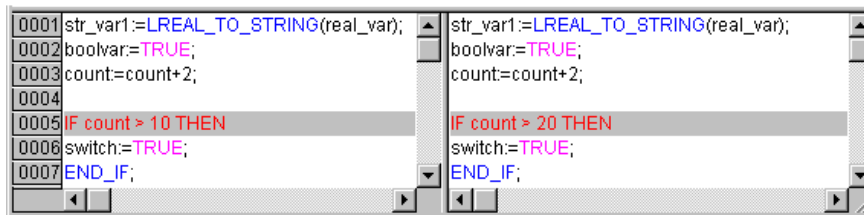
Compare differences: If a line, a network or an element within a POU has been modified, in compare mode it will be displayed in the bipartite window directly opposite to the version of the other project (marked red, see below). If the option is deactivated, the corresponding line will be displayed in the reference project as 'deleted' and in the actual project as 'inserted' (blue/green, see below). This means it will not be displayed directly opposite to the same line in the other project.

Example:

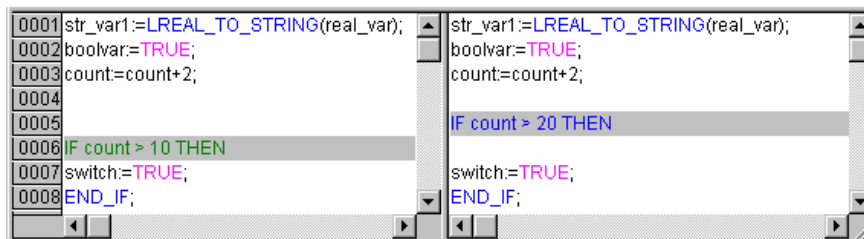
Line 0005 has been modified in actual project (left side).

Example for "Oppose differences"

Option 'Oppose differences' activated:



Option 'Oppose differences' not activated:



When the dialog 'Project Comparison' is closed by pressing OK, the comparison will be executed according to the settings.

Representation of the comparison result

First the structure tree of the project, titled with "Project Comparison", will be opened to display the results of the comparison. Here you can select particular POUs to see the found differences in detail.

1. Project overview in compare mode:

After the project compare has been executed, a bipartite window opens which shows the project in compare mode. In the title bar you find the project paths: "Project comparison <path of actual project> - <path of reference project>". The actual project is represented in the left half of the window, the reference project in the right one. Each structure tree shows the projects' name at the uppermost position, apart from that it corresponds to the object organizer structure.

POUs which are different, are marked in the structure tree by a shadow, a specific color and eventually by an additional text :

Red: Unit has been modified; is displayed with red colored letters in both partitions of the window.

Blue: Unit only available in compare project; a gap will be inserted at the corresponding place in the structure overview of the actual project.

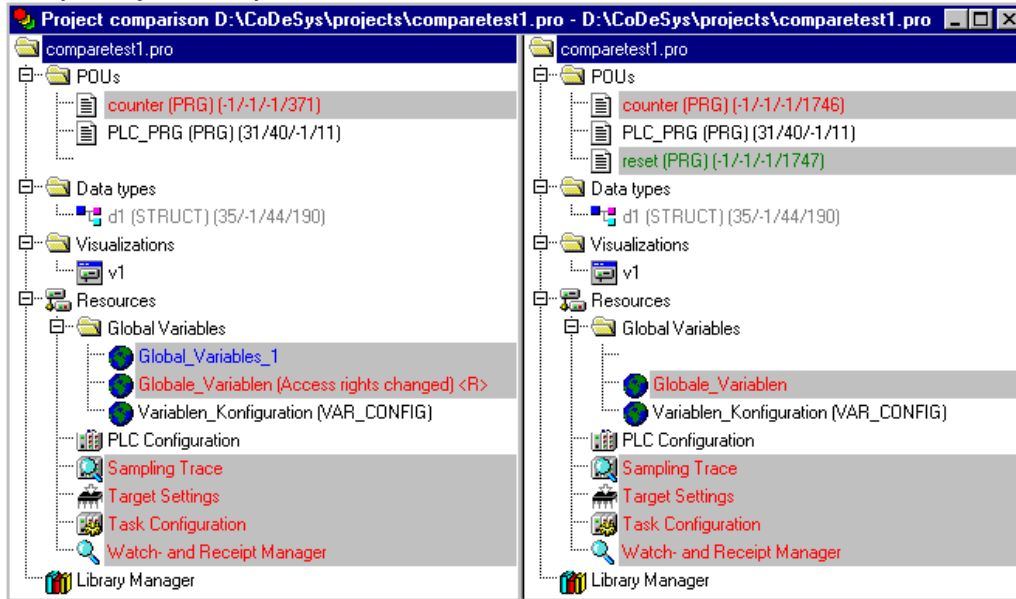
Green: Unit only available in actual project; a gap will be inserted at the corresponding place in the structure overview of the actual project.

Black: Unit for which no differences have been detected.

"(Properties changed)": This text is attached to the POU name in the project structure tree, if differences in the properties of the POU have been detected.

"(Access rights changed)": This text is attached to the POU name in the project structure tree, if differences in the access rights of the POU have been detected.

Example, Project in Compare mode

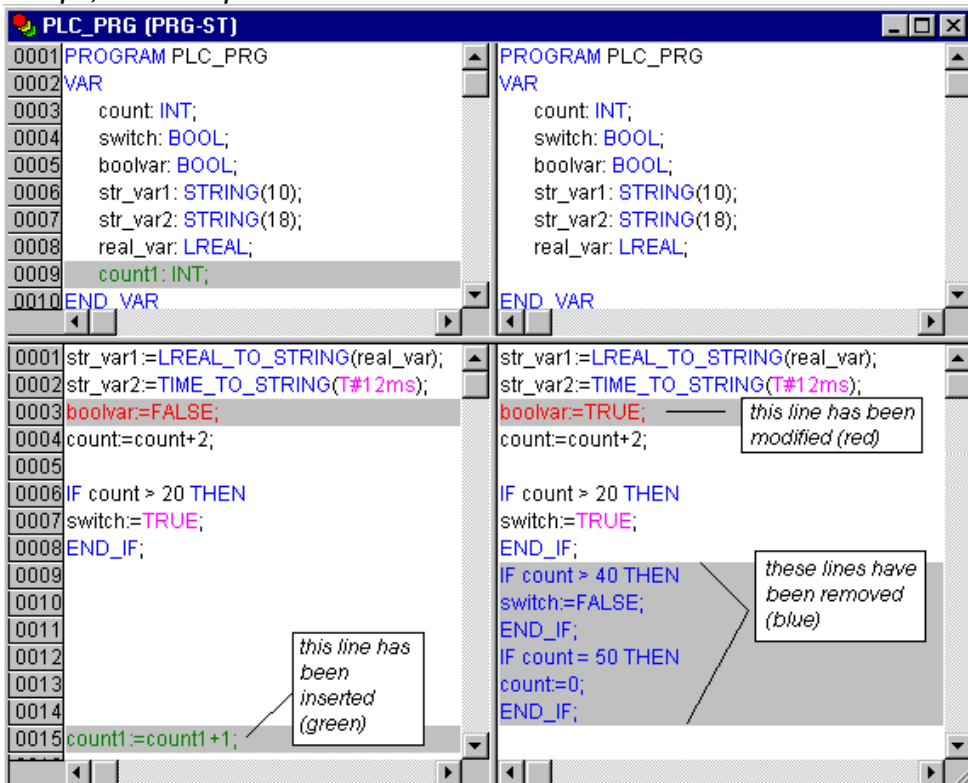


2. POU contents in compare mode:

By a double click on a line in the structure overview, which is marked red because of a modification, the POU is opened.

If it is a text or graphic editor POU, it will be opened in a bipartite window. The content of the reference project (right side) is set opposite to that of the actual project (left side). The smallest unit which will be regarded during comparison is a line (declaration editor, ST, IL), a network (FBD, LD) or an element (CFC, SFC). The same coloring will be used as described above for the project overview.

Example, POU in compare mode:



If it is not a editor POU, but the task configuration, the target settings etc., then the POU version of the actual and the reference project can be opened in separate windows by a double click on the respective line in the project structure. For those project POUs no further details of differences will be displayed.

Working in the compare mode

If in the bipartite window the cursor is placed on a line, which indicates a difference, the menu 'Extras' resp. The context menu offers a selection of the following commands, depending on whether working in the project overview or in a POU.

Note: The acceptance of different project parts (differences) or access right properties is only possible from the reference project of the actual project, not vice versa.

'Extras' 'Next difference'

Shortcut: <F7>

This command is available in the compare mode (see above 'Project' 'Compare').

The cursor jumps to the next unit, where a difference is indicated. (line in project overview, line/network/element in POU)

'Extras' 'Previous difference'

Shortcut: <Shift><F7>

This command is available in the compare mode (see above 'Project' 'Compare').

The cursor jumps to the previous unit, where a difference is indicated. (line in project overview, line/network/element in POU)

'Extras' 'Accept change'

Shortcut: <Space>

This command is available in the compare mode (see above 'Project' 'Compare').

For all units, which are cohering with that one where the cursor currently is placed, and which have the same sort of difference marking, (e.g. subsequent lines), the version of the reference project will be accepted for the actual project (only possible in this direction!). The corresponding units will be shown (with the corresponding coloring) in the left side of the window.

For accepting changes of particular units please use 'Accept changed item'.

'Extras' 'Accept changed item'

Shortcut: <Ctrl> <Spacebar>

This command is available in the compare mode (see above 'Project' 'Compare').

Only the single unit (line, network, element) where the cursor is currently placed, will be accepted for the actual version. The corresponding units will be shown (with the corresponding coloring) in the left side of the window.

If for a POU, which has got marked red-colored in the structure tree because of a change of its content, this change gets accepted, then the POU will be indicated by yellow coloring in the actual project. POUs which are only available in the current project because of 'Accept changed item', also will be indicated by yellow coloring. POUs which have been removed from the current project because of 'Accept changed item', will be indicated by yellow coloring in the reference project.

'Extras' 'Accept properties'

This command is available in the compare mode (see above 'Project' 'Compare').

The object properties for the POU, where the cursor is currently placed, will be accepted for the actual project as they are set in the reference version (acceptance only possible in this direction!).

'Extras' 'Accept access rights'

This command is available in the compare mode only in project overview: (see above 'Project' 'Compare').

The object access rights for the POU, where the cursor is currently placed, will be accepted for the actual project as they are set in the reference version .

'Project' 'Merge'

With this command you can merge objects (POUs, data types, visualizations, and resources) as well as links to libraries from other projects into your project.

When the command has been given, first the standard dialog box for opening files appears. When you have chosen a file there, a dialog box appears in which you can choose the desired object. The selection takes place as described with 'Project' 'Document' .

If an object with the same name already exists in the project, then the name of the new object receives the addition of an underline and a digit ("_1", "_2" ...).

'Project' 'Project info'

Under this menu item the information about your project can be saved. When the command has been given, then the dialog box 'Project Info' opens.

The following project information is displayed:

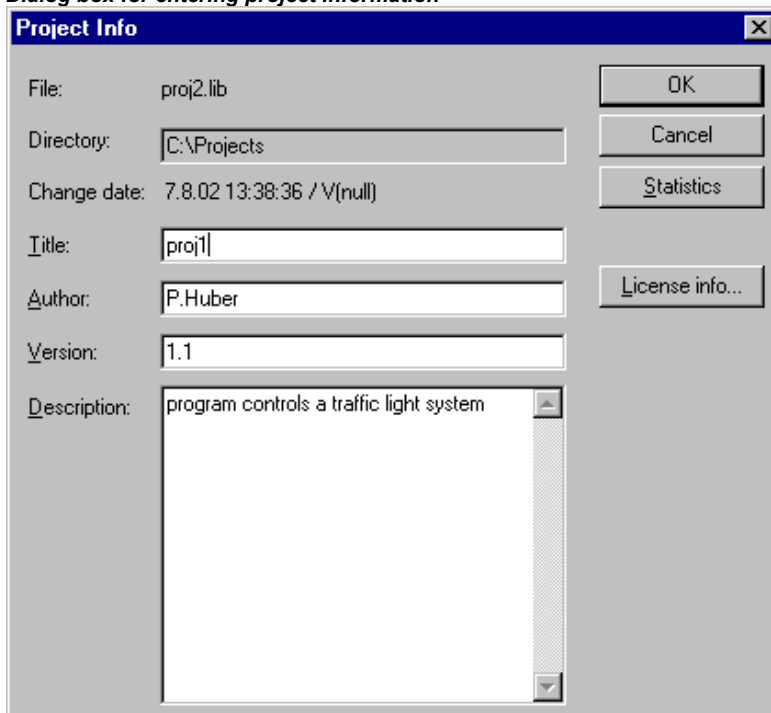
File name

Directory path

The time of the most recent change (**Change date**)

This information can not be changed.

Dialog box for entering project information



In addition, you can add the following information:

A **Title** of the project:

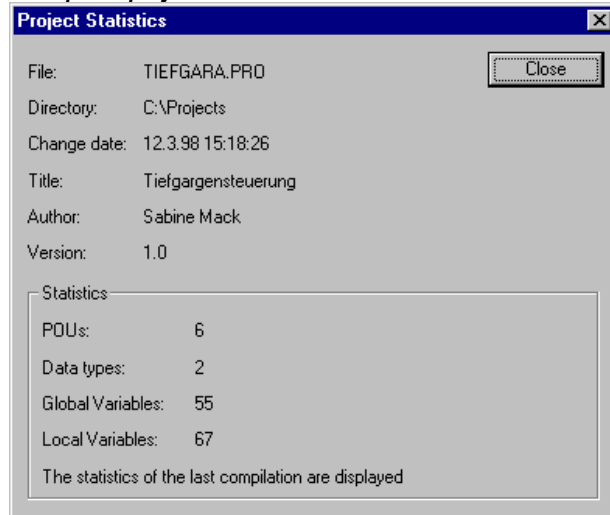
Please regard: If supported by the target system, this title automatically will be proposed as project file name, when the project gets loaded by command 'File' 'Open project from PLC' (In this case the dialog for saving a file will open).

the name of the **Author**,
 the **Version** number, and
Description of the project.

This information is optional. When you press the button **Statistics** you receive statistical information about the project.

It contains information such as the number of the POU's, data types, and the local and global variables as they were traced at the last compilation.

Example of project statistics



The button **License info** will be available, if you work on a CoDeSys project, which had been saved already with licensing information by the command 'File' 'Save as...'. In this case the button opens the dialog 'Edit Licensing Information', where you can modify or remove the license (see Chapter 9, 'License Management in CoDeSys')


If you choose the option **Ask for project info** in the category **Load & Save** in the Options dialog box, then while saving a new project, or while saving a project under a new name, the project info dialog is called automatically.

'Project' 'Global Search'

With this command you can search for the location of a text in POU's, data types, or in the objects of the global variables.

When the command is entered, a dialog box opens in which you can choose the desired object. The selection is made as in the 'Project' 'Document' description.

If the selection is confirmed with **OK**, the standard dialog for Search will be opened. This appears

immediately when the command 'Global Search' is invoked via the symbol  in the menu bar; the search is then automatically carried out in all searchable parts of the project. The most recently entered search strings can be selected through the combo box of the **Search for** field. If a text string is found in an object, the object is loaded into the corresponding editor or in the library manager and the location where the string was found is displayed. The display of the text that is found, as well as the search and find next functions behave similarly to the command 'Edit' 'Search'.

If you select the **In message window** button, all locations where the series of symbols searched for appears in the selected object will be listed line by line in tabular form in the message window. Afterward, the number of locations found will be displayed.

If the report window was not opened, it will be displayed. For each location that is found, the following will be displayed:

Object name

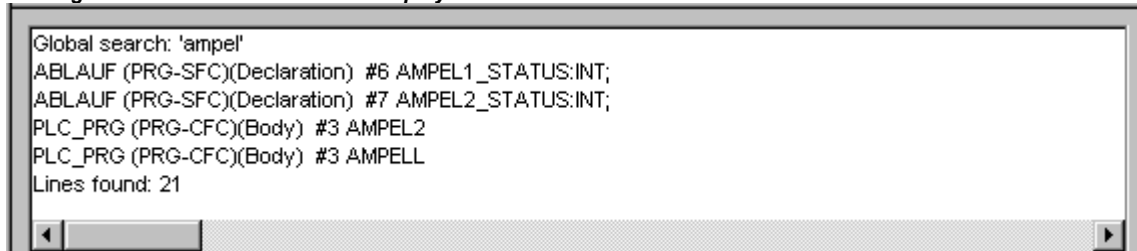
Location of the find in the Declaration (Decl) or in the Implementation (Impl) portion of a POU

Line and network number if any

The full line in the text editors

Complete text element in the graphic editors

Message window with search result display



If you double-click the mouse on a line in the message window or press <Enter>, the editor opens with the object loaded. The line concerned in the object is marked. You can jump rapidly between display lines using the function keys <F4> and <Shift>+<F4>.

'Project' 'Global Replace'

With this command you can search for the location of a text in POU's, data types, or the objects of the global variables and replace this text by another. This is executed in the same way as with 'Project' 'Global Search' or 'Edit' 'Replace'. The libraries, however, are not offered for selection and no display in the message window is possible.

Results are displayed in the message window.

'Project' 'Check'

This command provides commands for checking the semantic correctness of the project. The status of the most recent compilation will be regarded. In order to get up to date check results, you should recompile the project after any changes. An appropriate warning will be displayed in the message window.

A submenu listing the following commands will open:

Unused Variables

Overlapping memory areas

Concurrent Access

Multiple Write Access on Output

The results will be displayed in the message window.

Please regard: In the project options, category 'Build', you can define these semantic checks to be done at each compilation of the project automatically.

Unused Variables

This function in the 'Project' 'Check' menu (see above) searches for variables that have been declared but not used in the program. They are outputted by POU name and line, e.g.: PLC_PRG (4) – var1. Variables in libraries are not examined.

Results are displayed in the message window.

Overlapping memory areas

This function in the 'Project' 'Check' menu (see above) tests whether in allocation of variables via the "AT" declaration overlaps have arisen at specific memory areas. For example, an overlap occurs when allocating the variables "var1 AT %QB21: INT" and "var2 AT %QD5: DWORD" because they both use byte 21. The output then appears as follows:

%QB21 is referenced by the following variables:

PLC_PRG (3): var1 AT %QB21

PLC_PRG (7): var2 AT %QD5

Results are displayed in the message window.

Multiple Write Access on Output

This function of the 'Project' 'Check' menu (see above) searches for memory areas to which a single project gains write access at more than one place. The output then appears as follows:

```
%QB24 is written to at the following locations:
PLC_PRG (3): %QB24
PLC_PRG.POU1 (8): %QB24
```

Results are displayed in the message window.

Concurrent Access

This function in the 'Project' 'Check' menu (see above) searches for memory areas of IEC addresses which are referenced in more than one task. No distinction is made here between read and write access. The output is for example:

```
%MB28 is referenced in the following tasks :
Task1 - PLC_PRG (6): %MB28 [read-only access]
Task2 - POU1.ACTION (1) %MB28 [write access]
```

Results are displayed in the message window.

User groups

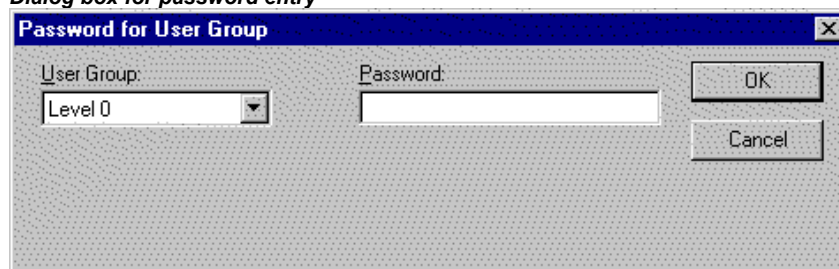
In CoDeSys up to eight user groups with different access rights to the POU's, data types, visualizations, and resources can be set up. Access rights for single objects or all of them can be established. Only a member of a certain user group can open a project. A member of such a user group must identify himself by means of a password.

The user groups are numbered from 0 to 7, whereby the Group 0 has the administrator rights, i.e. only members of group 0 may determine passwords and access rights for all groups and/or objects.

When a new project is launched, then all passwords are initially empty. Until a password has been set for the 0 group, one enters the project automatically as a member of the 0 group.

If a password for the user group 0 is existing while the project is loaded, then a password will be demanded *for all* groups when the project is opened. For this the following dialog box appears:

Dialog box for password entry



In the combo box **User group** on the left side of the dialog box, enter the group to which you belong and enter on the right side the relevant **password**. Press **OK**. If the password does not agree with the saved password, then the message appears:

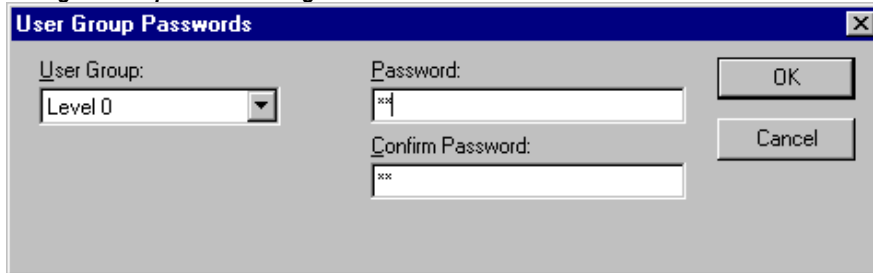
"The password is not correct."

Only when you have entered the correct password the project can be opened.

With the command 'Passwords for user group' you can assign the passwords, and with 'Object' 'Access rights' you can define the rights for single objects or for all of them.

'Project' 'User group passwords'

With this command you open the dialog box for password assignment for user groups. This command can only be executed by members of group 0. When the command has been given, then the following dialog box appears:

Dialog box for password assignment

In the left combo box **User group** you can select the group. Enter the desired password for the group in the field **Password**. For each typed character an asterisk (*) appears in the field. You must repeat the same password in the field **Confirm password**. Close the dialog box after each password entry with **OK**. If you get the message:

"The password does not agree with the confirmation",

then you made a typing error during one of the two entries. In this case repeat both entries until the dialog box closes without a message.

Then, if necessary, assign a password for the next group by calling the command again.

Important: If passwords are not assigned to all user groups, a project can be opened by way of a group to which no password was assigned!

Use the command '**Object' 'Access rights'** to assign the rights for single objects or all of them.

Concerning the protection of a project regard also the following:

- Passwords for access and write protection (Chapter 4.2, Options for Passwords)
- Encryption of a project at project saving (Chapter 4.3, 'File', 'Save as').

'Project' 'Data Base Link'

This menu item is only available if you have activated the option 'Use source control (ENI)' in the project options dialog for category 'Project source control'. A submenu is attached where you find the following commands for handling the object resp. the project in the currently connected ENI data base:

Login (The user logs in to the ENI Server)

If an object is marked in the Object Organizer and the command **Data Base Link** is executed (from the **context menu**, right mouse button), then the following commands will be available for executing the corresponding data base actions. If the user had not logged in successfully to the ENI Server before, then the dialog 'Data base Login' will open automatically and the chosen command will not be executed until the login was successful:

Define

Get Latest Version

Check Out

Check In

Undo Check Out

Show differences

Show Version History

If the command 'Data Base Link' in the 'Project' menu is activated, then additional menu items will be available, which concern all objects of the project:

- Multiple Define
- Get All Latest Versions
- Multiple Check Out
- Multiple Check In
- Multiple Undo Check Out
- Project Version History
- Label Version
- Add Shared Objects
- Refresh Status

See in the following how the status of an object resp. its handling in the data base is displayed in the Object Organizer :

	<p><u>Grey shaded icon:</u> Object is stored in the data base (source control)</p>
	<p><u>Green check in front of the object name:</u></p>
	<p>Object is checked out in the currently opened project.</p>
	<p><u>Red cross</u> in front of the object name: Object is currently checked out by another user.</p>
	<p><u><R></u> behind object name: The object can only be read, but not edited.</p>
	<p>Please regard: some objects (Task configuration, Sampling Trace, PLC Configuration, Target Settings, Watch-and Receipt Manager) are per default assigned with a <R> as long as they are not checked out. This means that you will not automatically be asked whether the object should be checked out, as soon as you start to edit the object; it not necessarily means that you cannot edit the object. If there is no write access then the command 'Check out' will not be available.</p>

Login

This command will open the dialog 'Login' where you can enter the access data for the ENI data base via the ENI Server. The access data also have to be defined in the ENI Server (ENI Admin, User Management) and – depending on the currently used data base – also in the user management of the data base. After the command has been executed, first the Login dialog for category 'Project objects' will open.

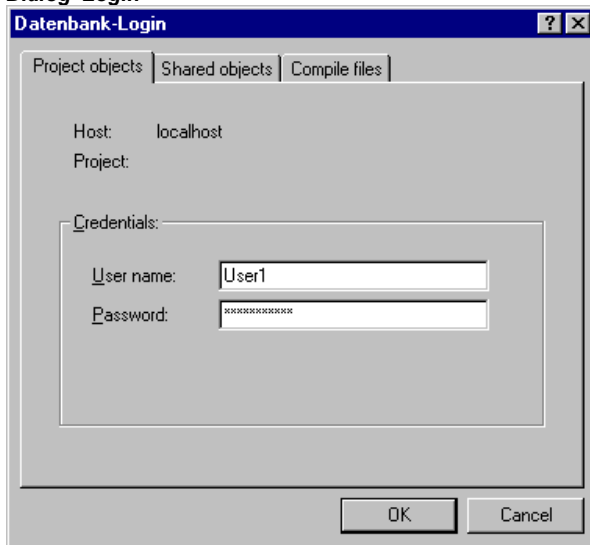
The following items are displayed:

Project objects:

Host: address of the computer where the ENI Server is running (must match with the entry in field 'TCP/IP address' in the project options dialog for 'Project source control').

Project: Name of the data base project (must match with the entry in field 'Project name' in the project options dialog for 'Project source control'/category 'Project Objects').

Dialog 'Login'



Credentials:

Insert **User name** and **Password**.

When option **Use as default** for this project is activated, then the above entered access data will automatically be used for any further communication between the actual CoDeSys project and the data base concerning objects of the actual category.

Press OK to confirm the settings. The dialog will be closed and automatically the Login dialog for 'Shared objects' will open. Enter the access data in the same way as described for the 'Project objects' and confirm with OK. Do the same in the third Login dialog which will be opened for category 'Compile files'.

The Login dialog will always open as soon as you try to access the data base before having logged in successfully like described above.

Note: If you want to save the access data with the project, activate option 'Save ENI credentials' in the project options, category 'Load & Save'.

Define

Command: 'Project' 'Data Base Link' 'Define'

Here you can define, whether the object which is currently marked in the Object organizer should be kept in the data base or just locally in the project. A dialog will open, where you can choose one of the two data base categories 'Project' or 'Shared objects', or the category 'Local'.

The icons of all objects which are managed in the data base will be displayed grey-shaded in the Object organizer.

Get Latest Version

Command: 'Project' Data Base Link'Get Latest Version'

The current version of the object which is marked in the Object organizer will be copied from the data base and will overwrite the local version. In contrast to the Check Out action the object will not be locked for other users in the data base.

Check Out

Command: 'Project' 'Data Base Link' 'Check Out'

The object which is marked in the Object organizer will be checked out from the data base and by that will be locked for other users.

When executing the command the user will get a dialog 'Check out object'. A comment can be added there which will be stored in the version history of the object in the data base. Line breaks are inserted by <Ctrl>+<Enter>. If the version of the object differs from that in the local project, an appropriate message will be displayed and the user can decide whether the object should be checked out anyway.

After the dialog has been closed with OK the checked-out object will be marked with a green check in the object organizer of the local project. For other users it will be appear marked with a red cross and will not be editable by them.

Check In

Command: 'Project' 'Data Base Link' 'Check In'

The object which is marked in the Object organizer will be checked in to the data base. Thereby a new version of the object will be created in the data base. The old versions will be kept anyway.

When executing the command the user will get a dialog 'Check in object'. There a comment can be added which will be stored in the version history of the object in the data base. Line breaks are inserted by <Ctrl>+<Enter>.

After the dialog has been closed with OK the green check in front of the object name in the Object organizer will be removed.

Undo Check Out

Command: 'Project' 'Data Base Link' 'Undo Check Out'

Use this command to cancel the Checking out of the object which is currently marked in the Object organizer. Thereby also the modifications of the object which have been made locally, will be cancelled. No dialog will appear. The unchanged last version of the object will be kept in the data base and it will be accessible again for other users. The red cross in front of the object name in the Object organizer will disappear.

Show Differences

Command: 'Project' 'Data Base Link' 'Show Differences'

The object which is currently opened by the user in CoDeSys will be displayed in a window which is divided up in two parts. There the local version, which is currently edited by the local user, will be opposed to the last (actual) version which is kept in the data base. The differences of the versions will be marked like described for the project comparison (see 'Project' 'Compare').

Show Version History

Command: 'Project' 'Data Base Link' Show Version History'

For the currently marked object in the Object organizer a dialog 'Version history of <object name> will be opened. There all versions of the object are listed which have been checked in to the data base or which have been labelled there.

The following information is given:

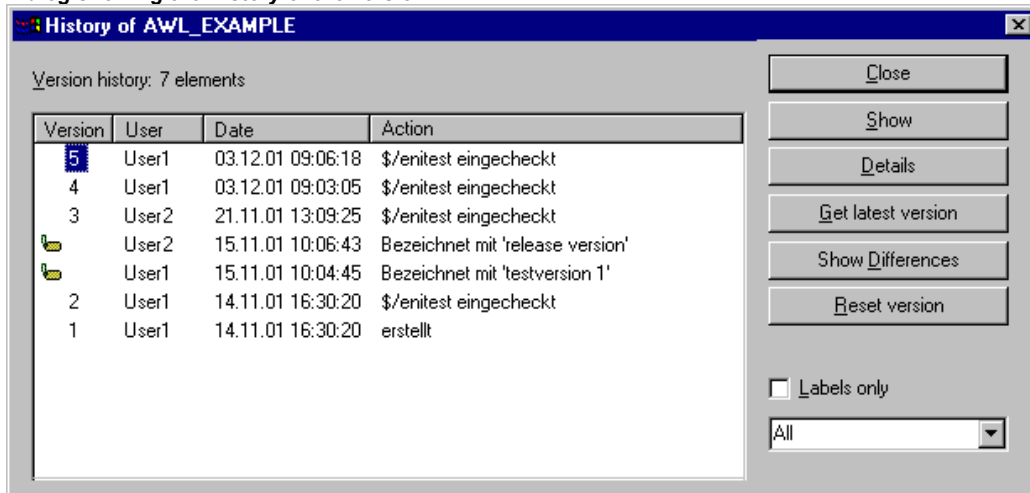
Version: Data base specific numbering of the versions of the object which have been checked in one after the other. Labelled versions get no version number but are marked by a label-icon.

User: Name of the user, who has executed the check-in or labelling action

Date: Date and time stamp of the action

Action: Type of the action which has been executed. Possible types: 'created' (the object has been checked in to the data base for the first time), 'checked in' (all check-in's of the object excluding the first one) and 'labeled with <label>' (a label has been assigned to this version of the object)

Dialog showing the History of the version



The buttons:

Close: The dialog will be closed.

Display: The version which is currently marked in the table will be opened in a window in CoDeSys. The title bar shows: "ENI: <name of the project in the data base>/<object name>

Details: The dialog 'Details of Version History' will open:

File (name of the project and the object in the data base), **Version** (see above), **Date** (see above), **User** (see above), Comment (Comment which has been inserted when the object has been checked in resp. has been labelled). Use the buttons **Next** resp. **Previous** to jump to the details window of the next or previous entry in the table in dialog 'Version history of ..'.

Get latest version: The version which is marked in the table will be loaded in CoDeSys and there will overwrite the local version.

Differences: If in the table only one version of an object is marked, then this command will cause a comparison of this version with the latest (actual) data base version. If two versions are marked, then those will be compared. The differences are displayed in a bipartite window like it is done at the project comparison.

Reset version: The version which is marked in the table will be set as latest version. All versions which have been checked in later will be deleted! This can be useful to restore an earlier status of an object.

Labels only: If this option is activated, then only those versions of the object will be displayed in the table, which are marked by a label.

Selection box below the option 'Labels only': Here you find the names of all users which have executed any data base actions for objects of the current project. Select 'All' or one of the names if you want to get the version history concerning all users or just for a certain one.

Multiple Define

Command 'Project' 'Data Base Link' 'Multiple Define'

Use this command if you want to assign several objects at a single blow to a certain data base category. The dialog '**Properties**' will open like described for command 'Define'. Choose the desired category and close the dialog with OK. After that the dialog '**ENI-Selection**' will open, listing all POUs of the project which are considered for the chosen category (Example: if you choose category 'shared objects' then the selection window will only offer the POUs of the Resources tab). The POUs are presented in a tree structure complying to that of the Object Organizer. Select the desired POUs and confirm with OK.

Get All Latest Versions

Command 'Project' 'Data Base Link' 'Get All Latest Versions'

The latest version of each object of the currently opened project, which is kept under source control, will be called from the data base. Consider the following:

If in the meantime additional objects have been stored to the data base project folder, then those will now be added to the local project in CoDeSys.

If objects have been deleted in the data base in the meantime, those will not be deleted in the local project, but they will automatically get assigned to category 'Local'.

The latest version of objects of category 'Shared Objects' will only be called, if these objects are already available in the local project. For further information see command 'Get latest version'.

Multiple Check Out

Command 'Project' 'Data Base Link' 'Multiple Check Out'

You can check out several objects at a single blow. For this the dialog '**ENI-Selection**' will open, listing all POUs of the project. Select those which should be checked out and confirm with OK. For further information see command 'Check Out'.

Multiple Check In

Command 'Project' 'Data Base Link' 'Multiple Check In'

You can check in several objects at a single blow. For this the dialog '**ENI-Selection**' will open, listing all POUs of the project. Select those which should be checked in and confirm with OK. For further information see command 'Check In'.

Multiple Undo Check Out

Command 'Project' 'Data Base Link' 'Undo Multiple Check Out'

You can undo the check out action for several objects at a single blow. For this the dialog '**ENI-Selection**' will open, listing all POUs of the project. Select those for which you want to cancel the check out and confirm with OK. For further information see command 'Undo Check Out'.

Project Version History

Command 'Project' 'Data Base Link' 'Project Version History'

If the chosen data base system supports that functionality, you can use this command to view the version history for the currently opened project.

The dialog 'History of <data base project name>' will open. It shows the actions (create, check in, label) which have been performed for the particular objects of the project in a chronological order. The total number of objects is displayed behind **Version history**. The dialog can be handled like described for command 'Show Version History', but regard the following:

The command 'Reset Version' is only available for single objects.

The command 'Get latest version' means that all objects of the version of the currently marked object will be called to the local project ! That means, that the objects in CoDeSys will be overwritten with the older version. But: Local objects, which were not yet part of the project in that older version, will

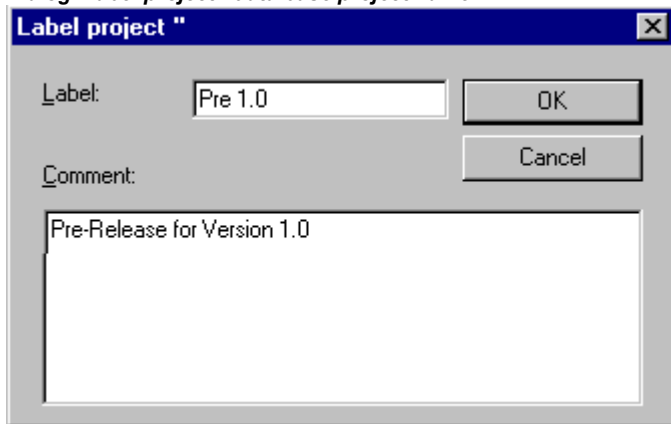
not be removed from the local project ! If a labeled version is called, which contains Shared Objects, then the user will get a dialog where he can decide whether those Shared Objects should be called also or not.

Label Version

Command 'Project' 'Data Base Link' 'Label Version'

This command is used to put a "label" on the actual version of each object of a project, so that exactly this project version can be recalled later. A dialog 'Label <data base project name>' will open. Insert a label name (**Label**) (e.g. "Release Version") and optionally a **Comment**. When you confirm with OK, the dialog will close and the label and the action "labeled with <label name>" will appear in the table of the version history, as well in the history for a single object as in the history of the project. Shared Objects which are part of the project will also get that label. A labeled version of the project does not get a version number, but is just marked with a label icon in the column 'Version'. If the option 'Labels only' is activated in the Version History dialog, then only labelled versions will be listed.

Dialog 'Label project <data base project name>'



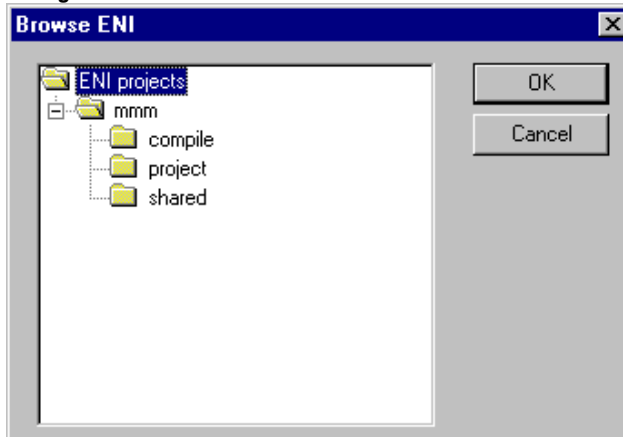
Add Shared Objects

Command 'Project' 'Data Base Link' 'Add Shared Objects'

Use this command if you explicitly want to add new objects of data base category 'Shared Objects' to the locally opened project in CoDeSys. For objects of category 'Project Objects' this is not necessary, because the command 'Get (all) latest version(s)' automatically calls all objects which are found in the data base project folder, even if there are some which not yet available in the local project. But for objects of category 'Shared Objects' in this case just those objects will be called which are already available in the local project.

So execute the command 'Add Shared Objects' to open the dialog 'Browse ENI'. A list in the right part of the window shows all objects which are available in the data base folder which is currently selected in the list on the left side. Choose the desired object and press OK or do a double-click on the entry to insert the object to the currently opened CoDeSys project.

Dialog 'Browse ENI'



Refresh Status

Command 'Project' 'Data Base Link' 'Refresh Status'

Use this command to update the display in the Object Organizer, so that you can see the actual status of the objects concerning the source control of the project.

4.4 Managing Objects in a Project...

In the following chapters find See book 'Managing information on how to work with objects and what help is available to keep track of a project (Folders, Call tree, Cross reference list,...).

Object


POUs, data types, visualizations and the resources global variables, the variable configuration, the Sampling Trace, the PLC Configuration, the Task Configuration, and the Watch and Receipt Manager are all defined as "objects". The folders inserted for structuring the project are partially involved. All objects of a project are in the Object Organizer.

If you hold the mouse pointer for a short time on a POU in the Object Organizer, then the type of the POU (Program, Function or Function block) is shown in a tooltip. For the global variables the tooltip shows the keyword (VAR_GLOBAL, VAR_CONFIG).

With drag & drop you can shift objects (and also folders, see 'Folder') within an object type. For this, select the object and shift it to the desired spot by holding down the left mouse button. If the shift results in a name collision, the newly introduced element will be uniquely identified by an appended, serial number (e.g. "Object_1").

Folder

In order to keep track of larger projects you should group your POUs, data types, visualizations, and global variables systematically in folders.

You can set up as many levels of folders as you want. If a plus sign is in front of a closed folder symbol , then this folder contains objects and/or additional folders. With a click on the plus sign the folder is opened and the subordinated objects appear. With a click on the minus (which has replaced the plus sign) the folder can be closed again. In the context menu you find the commands 'Expand nodes' and 'Collapse nodes' with the same functions.

With Drag&Drop you can move the objects as well as the folders within their object type. For this select the object and drag it with pressed left mouse button to the desired position.

You can create more folders with the command 'New folder'.

Note: Folders have no influence on the program, but rather serve only to structure your project clearly.

Example of folders in the Object Organizer



'New Folder'

With this command a new folder is inserted as a structural object. If a folder has been selected, then the new one is created underneath it. Otherwise it is created on the same level. If an action is selected, the new folder will be inserted at the level of the POU to which the action belongs.

The context menu of the Object Organizer which contains this command appears when an object or the object type has been selected and you have pressed the right mouse button or <Shift>+<F10>.

The newly inserted folder initially has the designation 'New Folder'. Observe the following naming convention for folders:

Folders at the same level in the hierarchy must have distinct names. Folders on different levels can have the same name.

A folder can not have the same name as an object located on the same level.

If there is already a folder with the name "New Folder" on the same level, each additional one with this name automatically receives an appended, serial number (e.g. "New Folder 1"). Renaming to a name that is already in use is not possible.

'Expand nodes' 'Collapse nodes'

With the command expand the objects are visibly unfolded which are located in the selected object. With Collapse the subordinated objects are no longer shown.

With folders you can open or close them with a double mouse click or by pressing <Enter>.

The context menu of the Object Organizer which contains this command appears when an object or the object type has been selected and you have pressed the right mouse button or <Shift>+<F10>.

'Project' 'Object Delete'

Shortcut: <Delete>

With this command the currently selected object (a POU, a data type, a visualization, or global variables), or a folder with the subordinated objects is removed from the Object Organizer and is thus deleted from the project. Deleting of an object can be reversed by the command 'Edit' 'Undo'.

You can get back the deleted objects by using the command 'Edit' 'Undo'.

If the editor window of the object was open, then it is automatically closed.

If you delete with the command 'Edit' 'Cut', then the object is parked on the clipboard.

'Project' 'Object Add'

Shortcut: <Insert>

With this command you create a new object. The type of the object (POU, data type, visualization, or global variables) depends upon the selected register card in the Object Organizer. Regard that in doing so possibly a template will be used for objects of type 'Global Variables', 'Data types', 'Function', 'Function Block' or 'Program' ., see below, chapter 'Save as template'.

Enter the **Name of the new POU** in the dialog box which appears. Remember that the name of the object may not have already been used.

Take note of the following restrictions:

The name of a POU can not include any spaces

A POU can not have the same name as another POU, a data type and should not have the same as a visualization in order to avoid problems with visualization changes.

A data type can not receive the same name as another data type or a POU.

A global variable list can not have the same name as another global variable list.

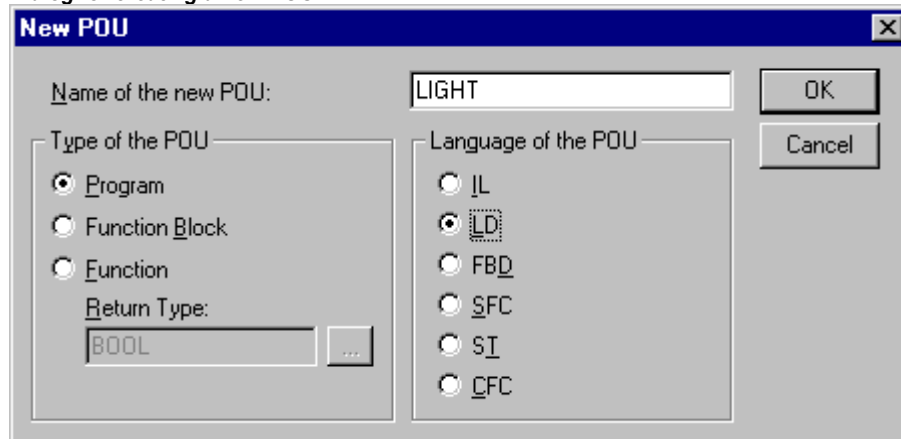
An action can not have the same name as another action in the same POU.

A visualization can not have the same name as another visualization and should not have the same as a POU in order to avoid problems with visualization changes.

In all other cases, identical naming is allowed. Thus for example actions belonging to different POUs can have the same name, and a visualization may have the same as a POU.

In the case of a POU, the POU type (program, function or function block) and the language in which it is programmed must also be selected. 'Program' is the default value of **Type of the POU**, while that of **Language of the POU** is that of most recently created POU. If a POU of the function type is created, the desired data type must be entered in the **Return Type** text input field. Here all elementary and defined data types (arrays, structures, enumerations, aliases) are allowed. Input assistance (e.g. via <F2>) can be used.

Dialog for creating a new POU



After pressing **OK**, which is only possible if there is no conflict with the naming conventions described above, the new object is set up in the Object Organizer and the appropriate input window appears.

If the command **'Edit' 'Insert'** is used, the object currently in the clipboard is inserted and no dialog appears. If the name of the inserted object conflicts with the naming conventions (see above), it is made unique by the addition of a serial number appended with a leading underline character (e.g. "Rightturnsig_1").

If the project is under source control in an ENI data base, it may be (depends on the settings in the Project options dialog for 'Project source control') that you will be automatically asked in which data base category you want to handle the new object. In this case the dialog 'Properties' will open where you can assign the object to one of the data base object categories.

'Save as template'

Objects of type 'Global Variables', 'Data types', 'Function', 'Function Block' or 'Program' can be saved as templates. Select the object in the Object Organizer and choose command 'Save as template' in the **context menu** (right mouse button). Hereupon each further new object of the same type will automatically initially get the declaration part of the template. The last created template for an object type will be used.

'Project' 'Object Rename'

Shortcut: <Spacebar>

With this command you give a new name to the currently-selected object or folder. Remember that the name of the object may not have already been used.

If the editing window of the object is open, then its title is changed automatically when the name is changed.

Dialog box for renaming a POU



'Project' 'Object Convert'

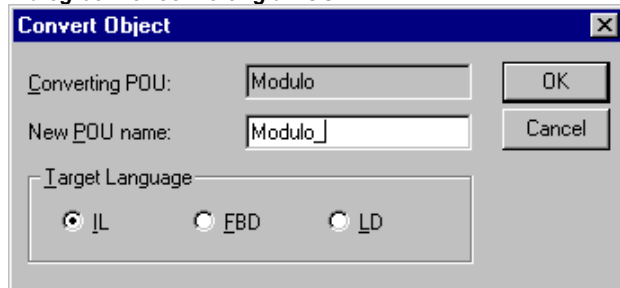
This command can only be used with POU's. You can convert POU's from the languages SFC, ST, FBD, LD, and IL into one of the three languages IL, FBD, and LD.

For this the project must be compiled. Choose the language into which you want to convert and give the POU a new name. Remember that the name of the POU may not have already been used. Then press **OK**, and the new POU is added to your POU list.

The type of processing that occurs during conversion corresponds to that which applies to compilation.

Attention: Actions cannot be converted.

Dialog box for converting a POU



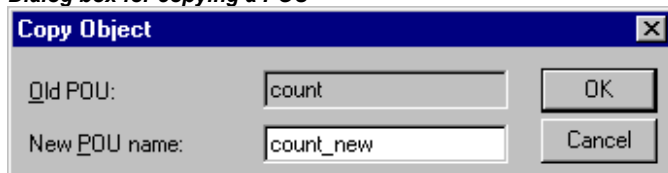
Regard the following possibility: A POU which has been created in the FBD-Editor, can - using the command 'Extras' 'View' be displayed and edited in the KOP-Editor as well without any conversion.

'Project' 'Object Copy'

With this command a selected object is copied and saved under a new name. Enter the name of the new object in the resulting dialog box. Remember that the name of the object may not have already been used.

If, on the other hand, you used the command 'Edit' 'Copy', then the object is parked on the clipboard, and no dialog box appears.

Dialog box for copying a POU



'Project' 'Object Open'

Shortcut: <Enter>

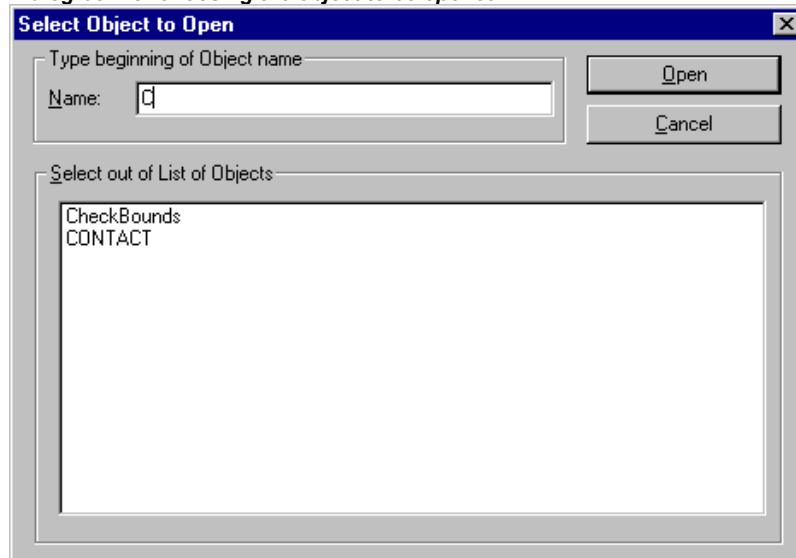
With the command you load a selected object within the Object Organizer into the respective editor. If a window with this object is already open, then it gets a focus, is moved into the foreground and can now be edited.

There are two other ways of opening an object:

Double-click with the mouse on the desired object

in the Object Organizer type the first letter of the object name. This will open a dialog box in which all objects of the available object types which have this initial letter are shown. Actions are listed with the notation <POU name>.<action name>. Due to the fact that the objects in the object selection dialog are listed alphabetically, the actions of a POU always get positioned below this POU. Select the desired object and click on the button **Open** in order to load the object in its edit window. Hereupon the object gets also marked in the object organizer and all folders which are hierarchically placed above the object will get expanded. This option is supported with the object type Resources only for global variables.

Dialog box for choosing the object to be opened



'Project' 'Object properties'

This command will open the dialog 'Properties' for that object which is currently marked in the Object organizer.

On the tab **Access rights** you find the same dialog as you get when executing the command 'Project' 'Object Access Rights'

It depends on the object and the project settings, whether there are additional tabs available where you can define object properties:

Global variable list:

In the tab 'Global variable list' the parameters concerning the actualization of the list and concerning the data exchange of network variables are displayed and can be modified here. This dialog also will be opened if you create a new global variable list by selecting one of the entries in section 'Global Variables' in the Object Organizer and executing the command 'Add Object'.

Visualization:

In the tab 'Visualization' you can define for the visualization object (see the CoDeSys Visualization User Manual), how it should be used:

Use as: If in the target settings the option 'Web-Visualization' resp. 'Target-Visualization' is activated, then you can choose here whether the object should be part of the Web-Visualization resp. Target-Visualization.

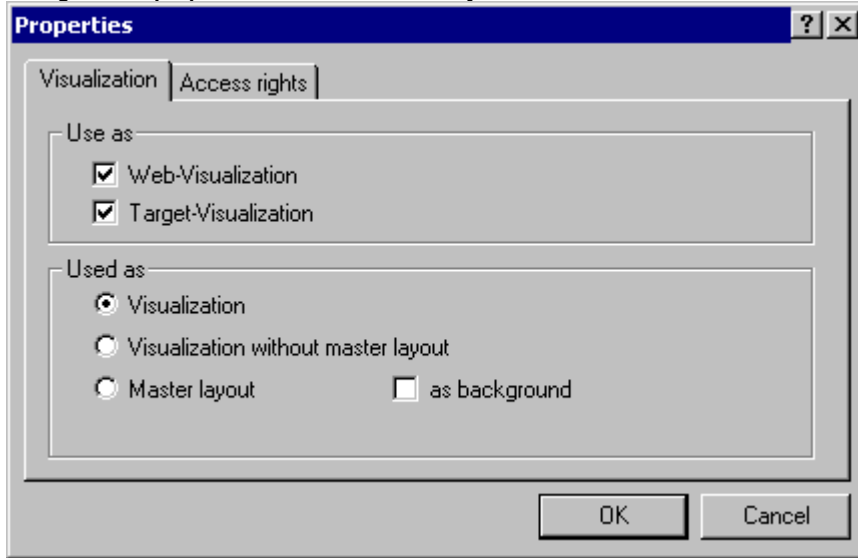
Used as: Activate one of these settings referring to the possibility of using "Master layouts":

Visualization: The object is used as a normal visualization.

Visualization without Master layout: : If a master layout is defined in the project, it will not be applied to this visualization object.

Master layout: The object will be used as master layout. Per default the master layout always will be in the foreground of a visualization, except the option **as background** is activated.

Dialog box for properties of a visualization object



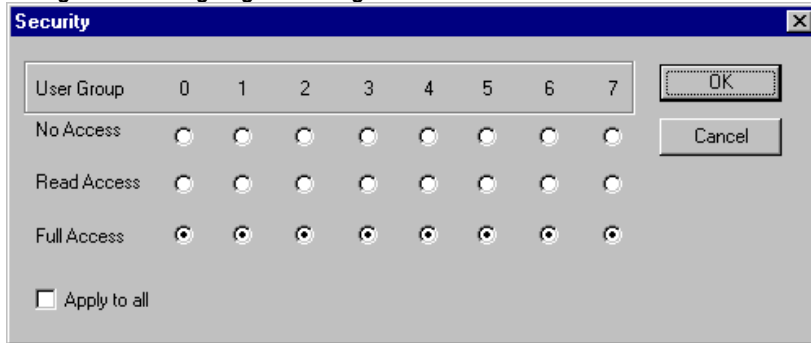
Database-connection:

If the project is connected to an ENI data base (see 'Project' 'Options' 'Project source control'), then a tab 'Database-connection' will be available. Here you can display and modify the current assignment of the object to one of the data base categories resp. to the category 'Local'. See for further information: 'What is ENI'.

'Project' 'Object Access rights'

With this command you open the dialog box for assigning access rights to the different user groups. The following dialog box appears:

Dialog box for assigning access rights



Members of the user group 0 can now assign individual access rights for each user group. There are three possible settings:

No Access: the object may not be opened by a member of the user group.

Read Access: the object can be opened for reading by a member of the user group but not changed.

Full Access: the object may be opened and changed by a member of the user group.

The settings refer either to the currently-selected object in the Object Organizer or, if the option **Apply to all** is chosen, to all POU's, data types, visualizations, and resources of the project.

The assignment to a user group takes place when opening the project through a password request if a password was assigned to the user group 0.

Please regard also the possibility to assign access rights concerning the operation of visualization elements (Visualization, Security).

'Project' 'Add Action'

This command is used to generate an action allocated to a selected block in the Object Organizer. One selects the name of the action in the dialog which appears and also the language in which the action should be implemented.

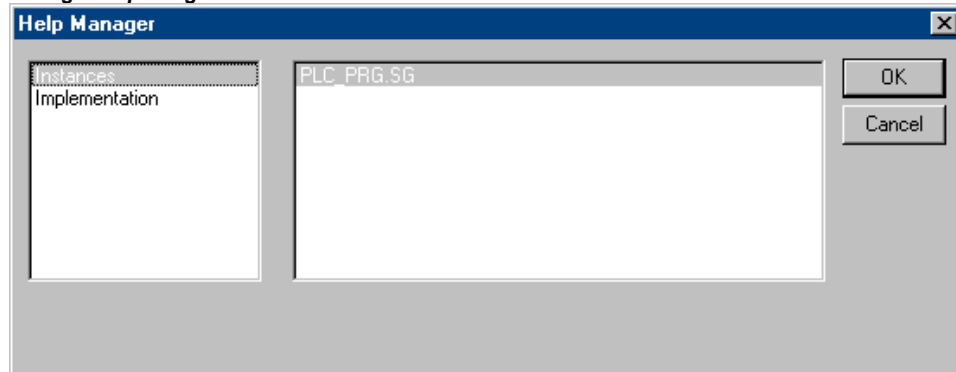
The new action is placed under your block in the Object Organizer. A plus sign appears in front of the block. A simple mouse click on the plus sign causes the action objects to appear and a minus sign appears in front of the block. Renewed clicking on the minus sign causes the actions to disappear and the plus sign appears again. This can also be achieved over the context menu commands 'Expand Node' and 'Collapse Node'.

'Project' 'View Instance'

With this command it is possible to open and display the instance of the function block which is selected in the Object Organizer. In the same manner, a double click on the function block in the Object Organizer gives access to a selection dialog in which the instances of the function block as well as the implementation are listed. Select here the desired instance or the implementation and confirm using OK. The desired item is then displayed in a window.

Please regard: If you want to view instances, you first have to log in ! (The project has been compiled with no errors and downloaded to the PLC with 'Online' 'Login').

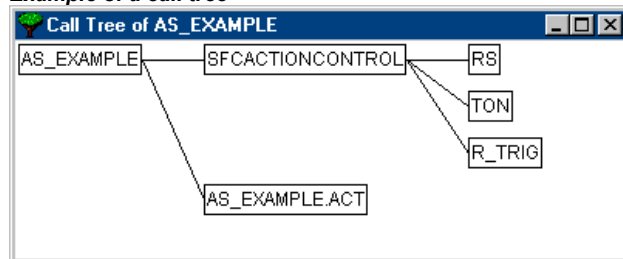
Dialog for opening an instance



'Project' 'Show Call Tree'

With this command you open a window which shows the call tree of the object chosen in the Object Organizer. Before this the project must have been compiled without any error (see 'Rebuild all'). The call tree contains both calls for POU's and references to data types.

Example of a call tree



'Project' 'Show cross reference'

With this command you open a dialog box which makes possible the output of all application points for a variable, address, or a POU. For this the project must be compiled (see 'Project' 'Build').

Choose first the **category** 'Variable', 'Address', or 'POU' and then enter the **name** of the desired element (the input assistant <F2> can be used). To obtain all elements of the entered category enter a "*" in Name.

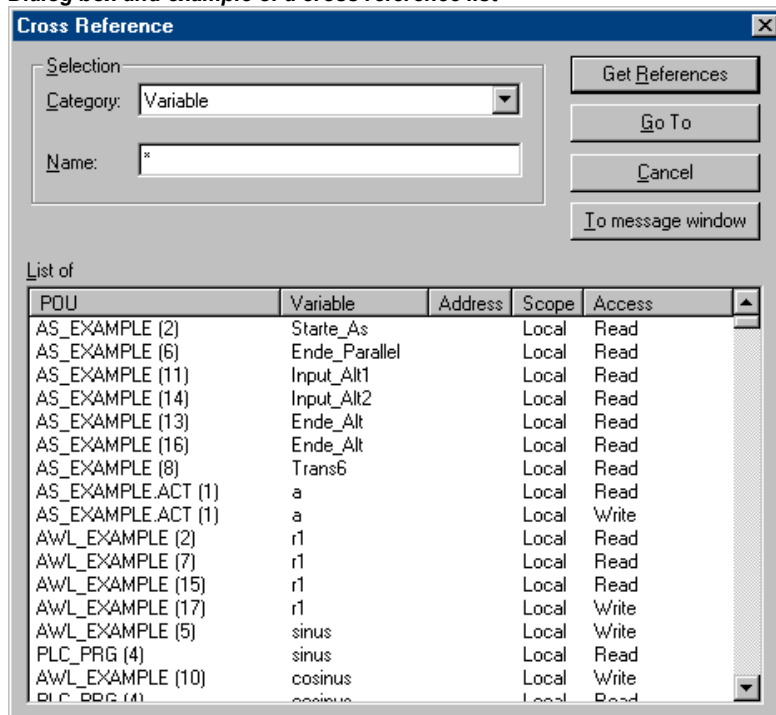
If the project has been changed since the last compile, the term "(Not up to date)" will be displayed in the title bar of the dialog. In this case any cross references which have been created recently will not be regarded in the list unless you do a re-compile !

By clicking on the button **Cross References** you get the list of all application points. Along with the POU and the line or network number, the variable name and the address binding, if any, are specified. The Domain space shows whether this is a local or a global variable; the Access column shows whether the variable is to be accessed for ,reading' or ,writing' at the current location. The column width will be adapted automatically to the entries' length.

When you select a line of the cross reference list and press the button **Go To** or double-click on the line, then the POU is shown in its editor at the corresponding point. In this way you can jump to all application points without a time-consuming search.

In order to make processing easier, you can use the **Send to message window** button to bring the current cross reference list into the message window and from there change to the respective POU.

Dialog box and example of a cross reference list



4.5 General Editing Functions...

You can use the following commands in all editors and some of them in the Object Organizer. The commands are located under the menu item **'Edit'** and in the context menu that is opened with the right mouse button.

If the IntelliPoint-Software is installed on the computer, CoDeSys supports all functions of the MS IntelliMouse. In all editors with zoom functionality: To magnify press the <Ctrl> key while rolling the wheel of the mouse, to reduce roll backwards while the <Ctrl> key is pressed.

'Edit' 'Undo'

Shortcut: <Ctrl>+<Z>

This command undoes the action which was most recently executed in the currently-open editor window or in the Object Organizer; repeated use undoes all actions back to the time that the window was opened. This applies to all actions in the editors for POUs, data types, visualizations and global variables and in the Object Organizer.

With 'Edit' 'Redo' you can restore an action which you have undone.

Note:	The commands Undo and Redo apply to the current window. Each window carries its own action list. If you want to undo actions in several windows, then you must activate the corresponding window. When undoing or redoing in the Object Organizer the focus must lie here.
--------------	--

'Edit' 'Redo'

Shortcut: <Ctrl>+<Y>

With the command in the currently-open editor window or in the Object Organizer you can restore an action you have undone ('Edit' 'Undo').

As often as you have previously executed the command '**Undo**', you can also carry out the command '**Redo**'.

Note:	The commands ' Undo ' and ' Redo ' apply to the current window. Each window carries its own action list. If you want to undo actions in several windows, then you must activate the corresponding window. When undoing or redoing in the Object Manager must lie there.
--------------	---

'Edit' 'Cut'

Symbol:  **Shortcut:** <Ctrl>+<X> or <Shift>+

This command transfers the current selection from the editor to the clipboard. The selection is removed from the editor.

In the Object Organizer this similarly applies to the selected object, whereby not all objects can be deleted, e.g. the PLC Configuration.

Remember that not all editors support the cut command, and that its use can be limited in some editors.

The form of the selection depends upon the respective editor:

In the text editors IL, ST, and declarations the selection is a list of characters.

In the FBD and LD editors the choice is a number of networks which are indicated by a dotted rectangle in the network number field or a box with all preceding lines, boxes, and operands.

In the SFC editor the selection is a part of a series of steps surrounded by a dotted rectangle.

In order to paste the content of the clipboard you use the command 'Edit' 'Paste'. In the SFC editor you can also use the commands 'Extras' 'Insert parallel branch (right)' or 'Extras' 'Paste after'.

In order to copy a selection onto the clipboard without deleting it, use the command 'Edit' 'Copy'.

In order to remove a selected area without changing the clipboard, use the command 'Edit' 'Delete'.

'Edit' 'Copy'

Symbol:  **Shortcut:** <Ctrl>+<C>

This command copies the current selection from the editor to the clipboard. This does not change the contents of the editor window.

With the Object Organizer this similarly applies to the selected object, whereby not all objects can be copied, e.g. the PLC Configuration.

Remember that not all editors support copying and that it can be limited with some editors.

For the type of selection the same rules apply as with '**Edit**' '**Cut**'.

The form of the selection depends upon the respective editor:

In the text editors (IL, ST, and declarations) the selection is a list of characters.

In the FBD and LD editors the choice is a number of networks which are indicated by a dotted rectangle in the network number field or a box with all preceding lines, boxes, and operands.

In the SFC editor the selection is a part of a series of steps surrounded by a dotted rectangle.

In order to paste the content of the clipboard you use the command 'Edit' 'Paste'. In the SFC editor you can also use the commands 'Extras' 'Insert parallel branch (right)' or 'Extras' 'Paste after'.

In order to delete a selected area and simultaneously put it on the clipboard, use the command 'Edit' 'Cut'.

'Edit' 'Paste'

Symbol:  **Shortcut:** <Ctrl>+<V>

Pastes the content of the clipboard onto the current position in the editor window. In the graphic editors the command can only be executed when a correct structure results from the insertion.

With the Object Organizer the object is pasted from the clipboard.

Remember that pasting is not supported by all editors and that its use can be limited in some editors.

The current position can be defined differently according to the type of editor:

With the text editors (IL, ST, Declarations) the current position is that of the blinking cursor (a vertical line) which you place by clicking with the mouse).

In the FBD and LD editors the current position is the first network with a dotted rectangle in the network number area. The contents of the clipboard are inserted in front of this network. If a partial structure has been copied, then it is inserted in front of the selected element.

In the SFC editor the current position is determined the selection which is surrounded by a dotted rectangle. Depending upon the selection and the contents of the clipboard, these contents are inserted either in front of the selection or into a new branch (parallel or alternative) to the left of the selection.

In SFC the commands 'Extras' 'Insert parallel branch (right)' or 'Extras' 'Paste after' can be used in order to insert the contents of the clipboard.

In order to copy a selection onto the clipboard without deleting it, use the command 'Edit' 'Copy'.

In order to remove a selected area without changing the clipboard, use the command 'Edit' 'Delete'.

'Edit' 'Delete'

Shortcut:

Deletes the selected area from the editor window. This does not change the contents of the clipboard.

In the Object Organizer this applies likewise to the selected object, whereby not all objects can be deleted, e.g. the PLC Configuration.

For the type of selection the same rules apply as with **'Edit' 'Cut'**.

The form of the selection depends upon the respective editor:

In the text editors (IL, ST, and declarations) the selection is a list of characters.

In the FBD and LD editors the selection is a number of networks which are highlighted with a dotted rectangle in the network number field.

In the SFC editor the selection is a part of a series of steps surrounded by a dotted rectangle.

In the library manager the selection is the currently selected library name.

In order to delete a selected area and simultaneously put it on the clipboard, use the command 'Edit' 'Cut'.

'Edit' 'Find'

Symbol: 

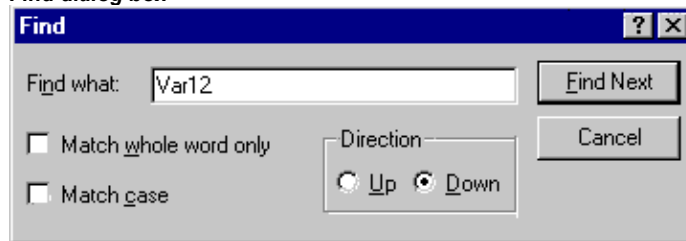
With this command you search for a certain text passage in the current editor window. The Find dialog box opens. It remains open until the button **Cancel** is pressed.

In the field **Find what** you can enter the series of characters you are looking for.

In addition, you can decide whether the text you are looking for **Match whole word only** or not, or also whether **Match case** is to be considered, and whether the search should proceed **Up** or **Down** starting from the current cursor position.

The button **Find next** starts the search which begins at the selected position and continues in the chosen search direction. If the text passage is found, then it is highlighted. If the passage is not found, then a message announces this. The search can be repeated several times in succession until the beginning or the end of the contents of the editor window has been reached. In the CFC editor the geometrical order of the elements will be regarded, the search will run from the left upper corner of the window to the right upper corner. Please regard that FBD POU's are processed from the right to the left !

Find dialog box

**'Edit' 'Find next'**

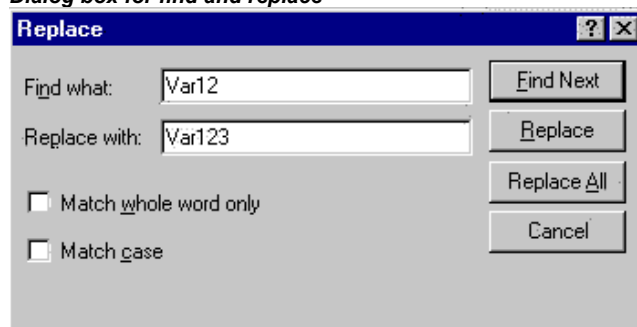
Symbol:  **Shortcut:** <F3>

With this command you execute a search with the same parameters as with the most recent action 'Edit' 'Find'. Please regard that FBD POU's are processed from the right to the left !

'Edit' 'Replace'

With this command you search for a certain passage just as with the command 'Edit' 'Find', and replace it with another. After you have chosen the command the dialog box for find and replace appears. This dialog box remains open until the button **Cancel** or **Close** is pressed.

Dialog box for find and replace



In the field behind **Find** automatically that string will be inserted which you have marked before in the editor. You also can enter the search string manually. Pressing button **Replace** will replace the current selection with the string which is given in the field **Replace with**. Use the button **Find Next** to get to the next passage where the string is found. Please regard, that FBD POU's are processed from the right to the left!

The button **Replace all** replaces every occurrence of the text in the field **Find next** after the current position with the text in the field **Replace with**. At the end of the procedure a message announces how many replacements were made.

'Edit' 'Input Assistant'

Shortcut: <F2>

This command provides a dialog box for choosing possible inputs at the current cursor position in the editor window. In the left column choose the desired input category, select the desired entry in the right column, and confirm your choice with **OK**. This inserts your choice at this position.

The categories offered depend upon the current cursor position in the editor window, i.e. upon that which can be entered at this point (e.g. variables, operators, POU's, conversions, etc.).

If the option **With arguments** is active, then when the selected element is inserted, the arguments to be transferred are specified with it, for example: function block fu1 selected, which defines the input variable var_in: fu1(var_in:=);

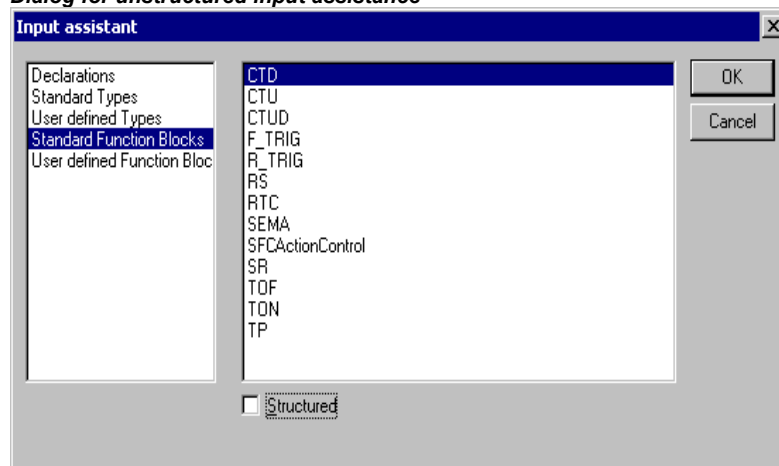
Insertion of function func1, which uses var1 and var2 as transfer parameters: func1(var1,var2)

It is basically possible to switch between structured and unstructured display of the available elements. This occurs through activation/deactivation of the **Structured Display** option.

Note: For inserting identifiers you also can use the "Intellisense functionality".

Unstructured Display

Dialog for unstructured input assistance



The POU's, variables or data types in each category are simply sorted linearly in alphabetical order.

At various places (e.g. in the Watch List), multi-stage variable names are required. In that event, the Input Assistant dialog displays a list of all POU's as well as a single point for the global variables. After each POU name there is a point. If a POU is selected by double-click or by pressing <Enter>, a list of the variables belonging to it opens. If instances and data types are present, it is possible to open further levels in the hierarchy display. **OK** transfers the most recently selected variable.

You can switch to structured display through activation of the **Structured Display**.

Structured Display

If **Structured** display is selected, the POU's, variables or data types will be sorted hierarchically. This is possible for standard programs, standard functions, standard function blocks, defined programs, defined functions, defined function blocks, global variables, local variables, defined types, watch variables. The visual and hierarchical display corresponds to that of the Object Organizer; if elements in a library are referred to, these are inserted in alphabetical order at the very top and the pertinent hierarchy is displayed as in the Library Manager.

The **in- and output variables** of function blocks which are declared as local or global variables are listed in the category 'Local Variables' or 'Global Variables' under the instance name (e.g. Inst_TP ET, Inst_TP IN,...). To get there, select the instance name (e.g. Inst_TP) and confirm with **OK**.

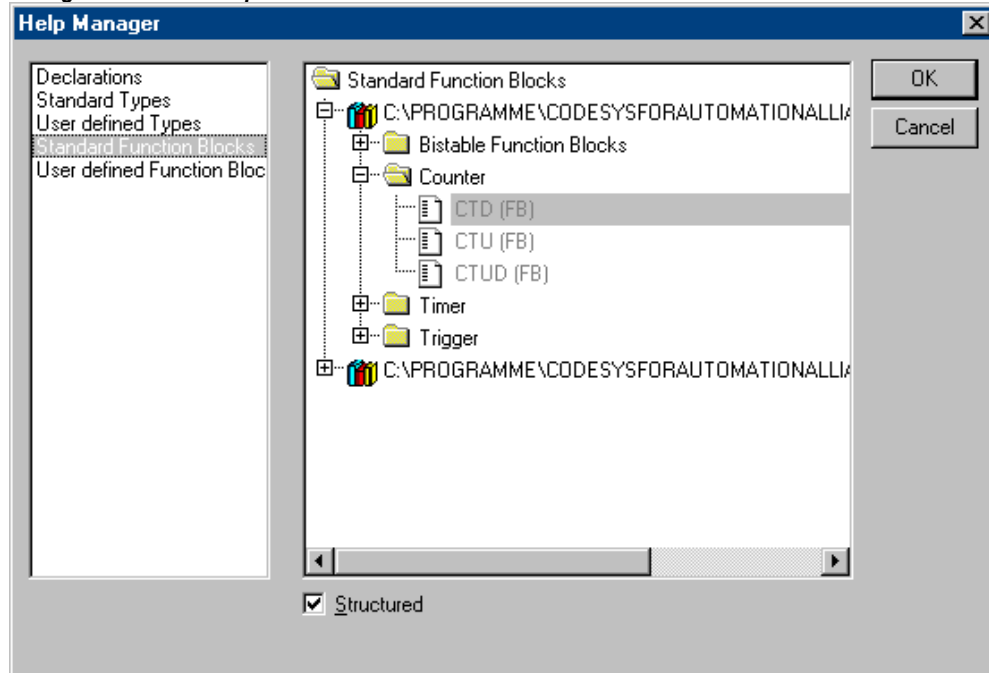
If the **instance of a function block** is selected here, the option **With arguments** may be selected. In the text languages ST and IL as well as during task configuration, the instance name and the input parameters of the function block are then inserted.

For example, if Inst (DeklarationInst: TON;) is selected, the following is inserted:

```
Inst(IN:= ,PT:=)
```

If the option is not selected, only the instance name will be inserted. In the graphical languages or in the Watch window, only the instance name is generally inserted.

Dialog for structured input assistance



Components of **structures** are displayed in an analogue fashion to function block instances.

For **enumerations**, the individual enumeration values are listed under the enumeration type. The order is: enumerations from libraries, enumerations from data types, local enumerations from POU's.

The general rule is that lines containing sub-objects are not selectable (except instances, see above), but can only have their hierarchy display expanded or contracted by one level, as for multi-stage variable names.

If Input Assistant is invoked in the Watch and Receipt Manager or in the selection of trace variables in the trace configuration dialog, it is possible to make a **multiple selection**. When the <Shift> key is pressed, you can select a range of variables; when the <Ctrl> key is pressed you can select many individual variables. The selected variables are so marked. If, during range selection lines are selected that do not contain valid variables (e.g. POU names), these lines will not be included in the selection. When individual selections are made, such lines can not be marked.

In the **watch window** and in **trace configuration** it is possible to transfer structures, arrays or instances from the Input Assistant dialog. As a double click with the mouse button is associated with the extension or contraction of the element's hierarchy display, selection in these cases can only be confirmed by **OK**.

Thereafter, the selected variables are inserted line by line in the watch window, that is each selected variable is written on a separate line. In the case of trace variables, each variable is inserted in a separate line of the trace variables list.

If the maximum number of trace variables, 20, is exceeded during insertion of the selected variables, the error message „A maximum of 20 variables is allowed" appears. Further selected variables are then not inserted in the list.

You can switch to unstructured display through deactivation of option **Structured**.

Note: Some entries (e.g. Global Variables) are only updated in the Input Assistant dialog after compilation.

'Edit' 'Autodeclare'

Shortcut: <Shift>+<F2>

This command opens the dialog for the declaration of a variable. This dialog also opens automatically when the option 'Project' 'Options' 'Editor' 'Autodeclaration' is switched on and when a new undefined variable is used the declaration editor.

'Edit' 'Next error'

Shortcut: <F4>

After the incorrect compilation of a project this command can show the next error. The corresponding editor window is activated and the incorrect place is selected. At the same time in the message window the corresponding error message is shown.

'Edit' 'Previous error'

Shortcut: <Shift>+<F4>

After the incorrect compilation of a project this command shows the previous error. The corresponding editor window is activated and the incorrect place is selected. At the same time in the message window the corresponding error message is shown.

'Edit' 'Macros'

This menu item leads to a list of all macros, which are defined for the project. (For info on generating macros see 'Project' 'Options' 'Macros'). When an executable macro is selected the dialog 'Process Macro' will open. The name of the macro and the currently active command line are displayed. The button Cancel can be used to stop the processing of the macro. In that event the processing of the current command will be finished anyway. Then an appropriate message is displayed in the message window and in the log during Online operation: "<Macro>: Execution interrupted by user".

Macros can be executed offline and online, but in each case only those commands are executed which are available in the respective mode.

4.6 General Online Functions...

The available online commands are assembled under the menu item '**Online**'. The execution of some of the commands depends upon the active editor.

The online commands become available only after logging in. See the following chapters for a description.

Thanks to 'Online Change' functionality you have the possibility of making changes to programs on the running controller. See in this connection 'Online' 'Login'.

See below in chapter 'Relations ...' a diagram showing the relations between Project-Build, Project-Download, Online Change and Login.

'Online' 'Login'

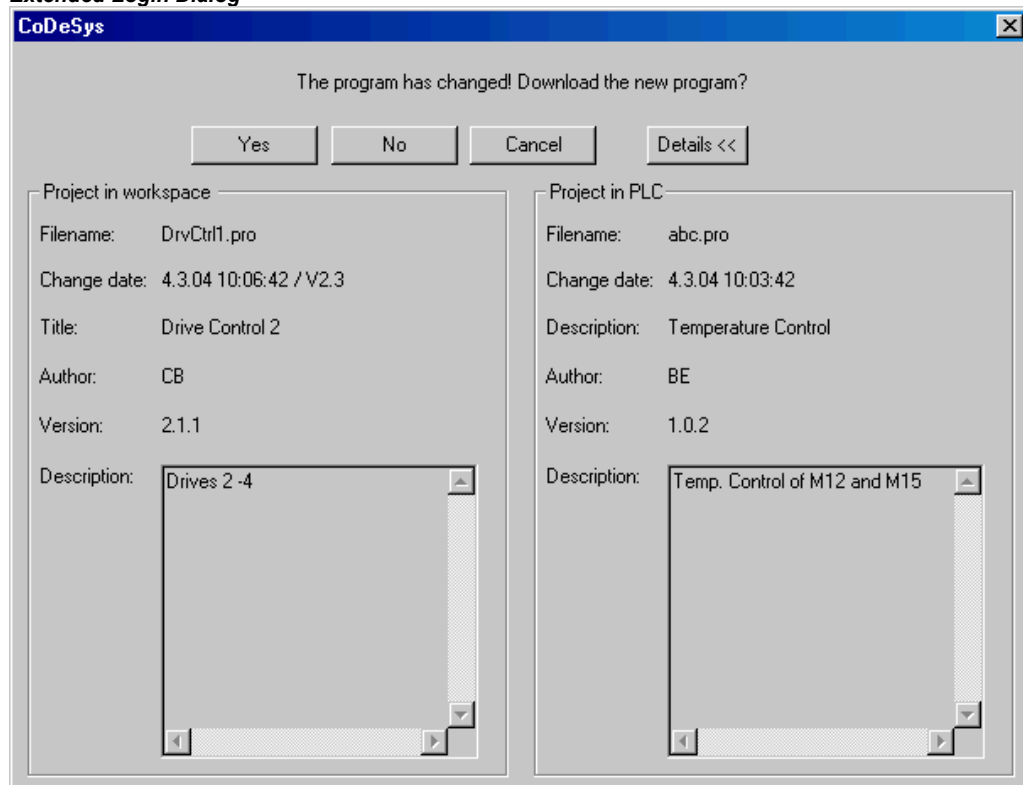
Symbol:  **Shortcut:** <Alt>+<F8>

This command combines the programming system with the PLC (or starts the simulation program) and changes into the online mode.

If the current project has not been compiled since opening or since the last modification, then it is compiled now (as with 'Project' 'Build'). If errors occur during compilation, then **CoDeSys** does not change into Online mode.

If the current project was changed on the controller since the last download, but not closed, and if the last download information was not deleted with the command 'Project' 'Clear all', then after the command 'Login' a dialog opens with the question: „The program has been changed. Load changes? (**Online Change**)". By answering **Yes** you confirm that, on log-in, the modified portions of the project are to be loaded onto the controller. (Concerning this matter see in the following chapters the hints on Online Change and a diagram showing the relations between Login, Compilation, Download and Online Change.) **No** results in a log-in without the changes made since the last download being loaded onto the controller. **Cancel** cancels the command. <Load all> causes the entire project to be reloaded onto the controller.

Extended Login Dialog



If in the project options, category Desktop, the option 'Online in security mode' is activated and if the target system supports the functionality, in the Login dialog automatically also the Project information will be displayed. This is the project information of the project which is currently opened in CoDeSys and which is already available on the controller. Via button **Details <<** you can close this information part of the dialog.

If the 'Online in security mode' option is not activated, you can explicitly open the project information display in the dialog via button **Details >>**.

Please regard that it depends on the target which button is set as default button.

Please regard: Online Change is not possible after modifications in the Task or PLC Configuration, after inserting a library and after performing 'Project' 'Clean all' (see below). Online Change does not cause a re-initialization of the variables, thus modifications of the initialization values will not be regarded! Retain variables keep their values when an Online Change is done, they won't do that at a re-download of the project (see below, 'Online' 'Download').

After a successful login all online functions are available (if the corresponding settings in 'Project' 'Options' category 'Build' have been entered). The current values are monitored for all visible variable declarations.

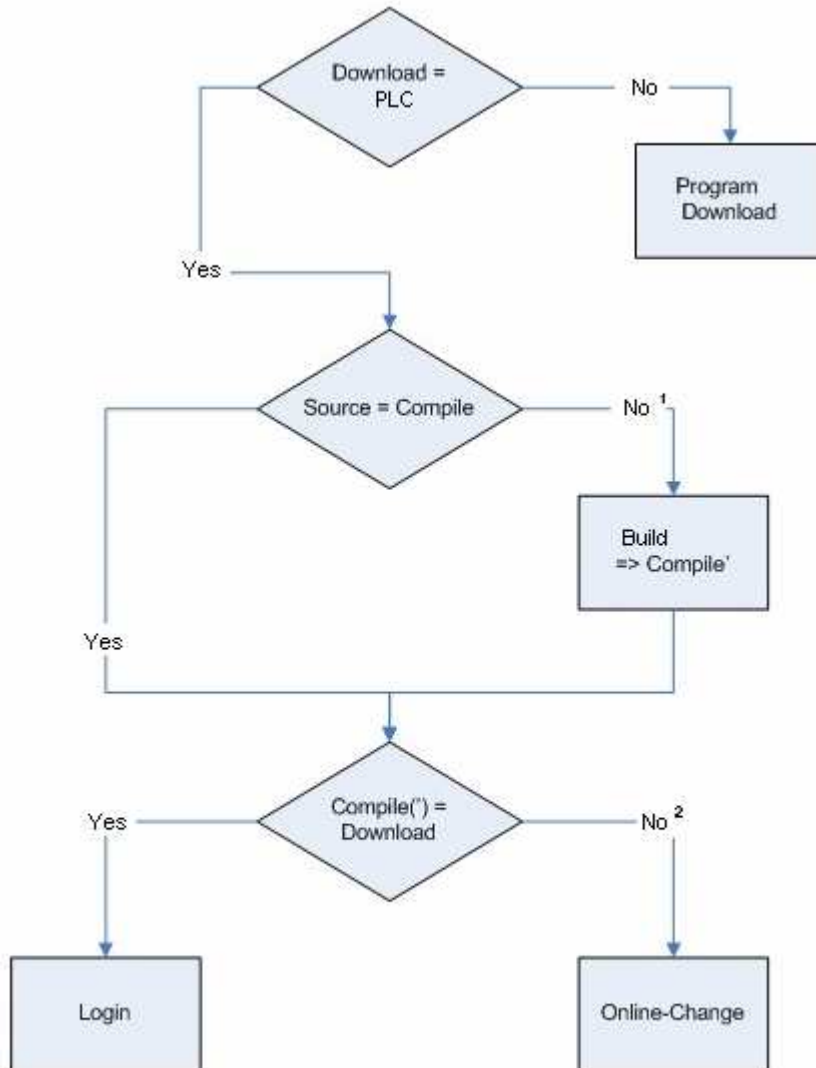
Use the 'Online' 'Logout' command to change from online back to offline mode.

Relations between Login – Build – Download – Online Change

See below a diagram which shows the relations between Login, Build (Compilation), Download and Online Change:

The following terms are used:

- Source current CoDeSys project (*.pro-file, local PC)
- Compile Compile information from the last build process, is needed for incremental compilation (*.ci-file, local PC)
- Download Information on what was loaded to the PLC at the last download (*.ri-file, local PC)
- PLC Project currently available on the PLC (*.prg-file, target system)



¹: e.g. changed libraries

²: e.g. changed compiler version or changed program code

Hints on Online Change

- Online Change is not possible after modifications in the Task or PLC Configuration, after inserting a library and after performing 'Project' 'Clean all' (see below).
- If the download Information (file <projectname><targetidentifier>.ri), which had been created at the last download (might have been an Online Change also) of the project, has been deleted meanwhile (e.g. via command 'Project' 'Clean all', then no Online Change will be possible further on, except for: The ri-file has been saved at another location or has been renamed and

therefore now is still available and can be loaded explicitly by command 'Project' 'Load download information'. Concerning this see also below 'Online Change for a project....'.

- Online Change does not cause a re-initialization of the variables, thus modifications of the initialization values will not be regarded !
- Retain variables keep their values when an Online Change is done, they won't do that at a re-download of the project (see below, 'Online' 'Download').

Online Change for a project which is running on several PLCs:

If you want to run a project *proj.pro* on two identical controllers PLC1 and PLC2 (same target system) and want to make sure that updates of the project on both controllers can be done via online change, do the following:

(1) Loading and starting project on PLC1, saving download information for PLC1:

1. Connect the CoDeSys project *proj.pro* to controller PLC1 (Online/Communication parameters) and load *proj.pro* on PLC1 (Online/Login, Download). At the download the file *proj00000001.ri* will be created in the projects directory, containing download information.
2. Rename *proj00000001.ri*, e.g. to *proj00000001_PLC1.ri*. This save of the file with another file name is necessary because at a further download of *proj.pro* the file *proj00000001.ri* on another PLC would be overwritten with new download information and thus the information belonging to the download on PLC1 would be lost.
3. Start the project on PLC1 and then log out ('Online' 'Start', 'Online' 'Logout').

(2) Loading and starting project on PLC2, saving download information for PLC2:

1. Now connect to controller PLC2 (using same target as PLC1) and download *proj.pro* on PLC2. Thus again a file *proj00000001.ri* will be created in the projects directory, now containing the information on the currently done download.
2. Rename the new *proj00000001.ri* e.g. to *proj00000001_PLC2.ri* in order to store it explicitly.
3. Start the project on PLC2 and log out ('Online' 'Start', 'Online' 'Logout').

(3) Modifying project in CoDeSys:

1. In CoDeSys now do the modifications in *proj.pro* which you afterwards want to transfer via Online Change to the program running on the both PLCs.

(4) Online Change on PLC1, Saving of the download information again for PLC1:

2. In order to make possible the Online Change for *proj.pro* on PLC1, first the download information referring to the download of *proj.pro* on PLC1 must be restored. At login CoDeSys is looking for a file *proj00000001.ri*. But you have stored the appropriate download information in file *proj00000001_PLC1.ri*. Now you have 2 possibilities:
 - (a) You can rename *proj00000001_PLC1.ri* again to *proj00000001.ri*. Thus at a login on PLC1 automatically the appropriate download information is available and CoDeSys will ask you whether you want to do an Online Change.
 - (b) Alternatively you can explicitly load file *proj00000001_PLC1.ri* via command 'Project' 'Load Download-Information', before logging in. Thereby you need not to rename the ri-file and nevertheless make possible an Online Change.
3. At the Online Change on PLC1 an updated version of file *proj00000001.ri* has been created, containing the current download information. Store this file again as described in (4), to keep it available for a further online change on PLC1.

(5) Online Change on PLC2, Saving of the download information again for PLC2:

In order to make possible an Online Change concerning the modifications in *proj.pro* done in (3) also on PLC2 please perform the corresponding steps for *proj00000001_PLC2.ri* as described in step (4).

(6) Each further Online Change after a project modification: Perform steps (3) to (5)

If the system reports

Error:

"The selected controller profile does not match that of the target system..."

Check that the target system entered in the target system settings (Resources) matches the parameters entered in 'Online' 'Communication parameters'.

Error:

„Communication error. Log-out has occurred"

Check whether the controller is running. Check whether the parameters entered in '**Online**' '**Communication parameters**' match those of your controller. In particular, you should check whether the correct port has been entered and whether the baud rates in the controller and the programming system match. If the gateway server is used, check whether the correct channel is set.

Error:

"The program has been modified! Should the new program be loaded?"

The project which is open in the editor is incompatible with the program currently found in the PLC (or with the Simulation Mode program being run). Monitoring and debugging is therefore not possible. You can either choose "No," logout, and open the right project, or use "Yes" to load the current project in the PLC.

Message:

„The program has been changed. Load changes? (ONLINE CHANGE)".

The project is running on the controller. The target system supports 'Online Change' and the project has been altered on the controller with respect to the most recent download or the most recent Online Change. You may now decide whether these changes should be loaded with the controller program running or whether the command should be cancelled. You can also, however, load the entire compiled code by selecting the **Load all** button.

'Online' 'Logout'

Symbol:  **Shortcut <Ctrl>+<F8>**

The connection to the PLC is broken, or, the Simulation Mode program is ended and is shifted to the offline mode.

Use the 'Online' 'Login' command to change to the online mode.

'Online' 'Download'

This command loads the compiled project in the PLC.

During compilation the download information gets saved in a file called **<projectname>0000000ar.ri** , which is used during Online Change to compare the current program with the one most recently loaded onto the controller, so that only changed program components are reloaded. This file is erased by the command 'Project' 'Clear all'. Concerning Online Change on several PLCs please see chapter 'Online' 'Login'. Regard that the *.ri-file also gets updated during an Online Change.

Depending on the target system settings at each creation of a boot project in offline mode the *.ri-file might be regenerated.

Only persistent variables (see Chapter 5.2.1, Remanent variables) keep their values even after a download.

'Online' 'Run'

Symbol:  **Shortcut: <F5>**

This command starts the program in the PLC or in Simulation Mode.

This command can be executed immediately after the 'Online' 'Download' command, or after the user program in the PLC has been ended with the 'Online' 'Stop' command, or when the user program is at a break point, or when the 'Online' 'Single Cycle' command has been executed.

'Online' 'Stop'

Symbol:  **Shortcut <Shift>+<F8>**

Stops the execution of the program in the PLC or in Simulation Mode between two cycles.

Use the 'Online' 'Run' command to continue the program.

'Online' 'Reset'

This command resets – with exception of the retain variables (VAR RETAIN) - all variables to that specific value, with which they have got initialized (also those variables which have been declared as VAR PERSISTENT !). If you have initialized the variables with a specific value, then this command will reset the variables to the initialized value. All other variables are set at a standard initialization (for example, integers at 0). As a precautionary measure, **CoDeSys** asks you to confirm your decision before all of the variables are overwritten. The situation is that which occurs in the event of a power failure or by turning the controller off, then on (warm restart) while the program is running.

Use the 'Online' 'Run' command to restart the program.

See also 'Online' 'Reset (original)', 'Online' 'Reset (cold)' and - for an overview on reinitialization – Chapter 5.2.1, Remanent variables.

'Online' 'Reset (cold)'

This command corresponds to the 'Reset' Command (see above) with the exception that also retain variables(!) are set back to their initialization values. The situation is that which occurs at the start of a program which has been downloaded just before to the PLC (cold start). Only persistent variables retain the value that they had before the reset. Referring to this see also 'Online' 'Reset', 'Online' 'Reset Original' and - for an overview on reinitialization - Chapter 5.2.1, Remanent variables.

'Online' 'Reset (original)'

This command resets all variables including the remanent ones (VAR RETAIN and VAR PERSISTENT) to their initialization values and erases the user program on the controller. The controller is returned to its original state. Referring to this see also 'Online' 'Reset', 'Online' 'Cold Reset' and - for an overview on reinitialization - Chapter 5.2.1, Remanent variables.

'Online' 'Toggle Breakpoint'

Symbol:  **Shortcut: <F9>**

This command sets a breakpoint in the present position in the active window. If a breakpoint has already been set in the present position, that breakpoint will be removed.

The position at which a breakpoint can be set depends on the language in which the POU in the active window is written.

In the Text Editors (IL, ST), the breakpoint is set at the line where the cursor is located, if this line is a breakpoint position (recognizable by the dark-gray color of the line number field). You can also click on the line number field to set or remove a breakpoint in the text editors.

In FBD and LD, the breakpoint is set at the currently selected network. In order to set or remove a breakpoint in the FBD or LD Editor, you can also click on the network number field.

In SFC, the breakpoint is set at the currently selected step. In SFC you can also use <Shift> with a double-click to set or remove a breakpoint.

If a breakpoint has been set, then the line number field or the network number field or the step will be displayed with a light-blue background color.

If a breakpoint is reached while the program is running, the program will stop, and the corresponding field will be displayed in a red background color. In order to continue the program, use the 'Online' 'Run', 'Online' 'Step in', or 'Online' 'Step Over' commands.

You can also use the Breakpoint dialog box to set or remove breakpoints.

'Online' 'Breakpoint Dialog Box'

This command opens a dialog box to edit breakpoints throughout the entire project. The dialog box also displays all breakpoints presently set.

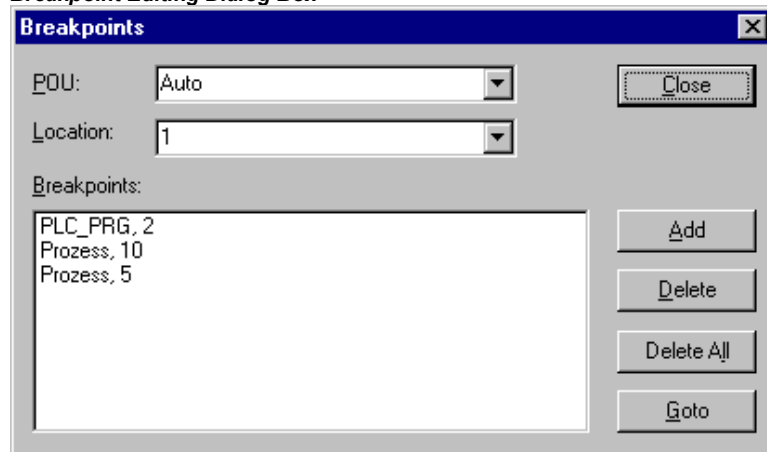
In order to set a breakpoint, choose a POU in the **POU** combo box and the line or the network in the **Location** combo box where you would like to set the breakpoint; then press the **Add** button. The breakpoint will be added to the list.

In order to delete a breakpoint, highlight the breakpoint to be deleted from the list of the set breakpoints and press the **Delete** button.

The **Delete All** button can be used to delete all the breakpoints.

In order to go to the location in the editor where a certain breakpoint was set, highlight the respective breakpoint from the list of set breakpoints and press the **Go to** button.

Breakpoint Editing Dialog Box



To set or delete breakpoints, you can also use the 'Online' 'Toggle Breakpoint' command.

'Online' 'Step over'

Symbol:  **Shortcut: <F10>**

This command causes a single step to execute. If a POU is called, the program stops after its execution. In SFC a complete action is executed.

If the present instruction is the call-up of a function or of a function block, then the function or function block will be executed completely. Use the 'Online' 'Step In' command, in order to move to the first instruction of a called function or function block.

If the last instruction has been reached, then the program will go on to the next instruction in the POU.

'Online' 'Step in'

Shortcut: <F8>

A single step is executed. The program is stopped before the first instruction of a called POU.

If necessary, there will be a changeover to an open POU.

If the present position is a call-up of a function or of a function block, then the command will proceed on to the first instruction in the called POU.

In all other situations, the command will function exactly as 'Online' 'Step Over'.

'Online' 'Single Cycle'**Shortcut: <Ctrl>+<F5>**

This command executes a single PLC Cycle and stops after this cycle.

This command can be repeated continuously in order to proceed in single cycles.

The Single Cycle ends when the 'Online' 'Run' command is executed.

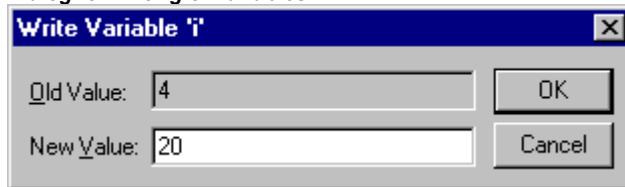
'Online' 'Write values'**Shortcut: <Ctrl>+<F7>**

With this command, one or more variables are set – one time only! – to user defined values at the beginning of a cycle. (see 'Online' 'Force values' for setting permanently)

The values of all single-element variables can be changed, so long as they are also visible in Monitoring.

Before the command 'Write values' can be executed, a variable value must be ready to be written:

For non-boolean variables a double mouse click is performed on the line in which a variable is declared, or the variable is marked and the <Enter> key is pressed. The dialog box 'Write variable <x>' then appears, in which the value to be written to the variable can be entered.

Dialog for writing of variables

For boolean variables, the value is toggled (switched between TRUE and FALSE, with no other value allowed) by double-clicking on the line in which the variable is declared; no dialog appears.

The value set for Writing is displayed in brackets and in turquoise colour behind the former value of the variable. e.g. a=0 <:=34>.

Hint: Exception: In the FBD and LD Editor the value is shown turquoise without brackets next to the variable name.

Set the values for as many variables as you like.

The values entered to be written to variables can also be corrected or deleted in the same manner. This is likewise possible in the 'Online' 'Write/Force dialog' (see below).

The values to be written that were previously noticed are saved in a **writelist (Watchlist)**, where they remain until they are actually written, deleted or transferred to a forcelist by the command 'Force values'.

The command to Write Values can be found at two places::

- Command 'Write Values' in the menu 'Online'.
- Button 'Write Values' in the dialog 'Editing the writelist and the forcelist'.

When the command 'Write values' is executed, all the values contained in the writelist are written, once only, to the appropriate variables in the controller at the beginning of the cycle, then deleted from the writelist. (If the command 'Force values' is executed, the variables in question are also deleted from the writelist, and transferred to the forcelist!)

Note: In the sequential function chart language (SFC), the individual values from which a transition expression is assembled cannot be changed with 'Write values'. This is due to the fact that in monitoring the 'Total value' of the expression, not the values of the individual variables are displayed (e.g. "a AND b" is only displayed as TRUE if both variables actually have the value TRUE).

In FBD, on the other hand, only the first variable in an expression, used for example as input to a function block, is monitored. Thus a 'Write values' command is only possible for this variable.

'Online' 'Force values'**Shortcut: <F7>**

With this command, one or more variables are permanently set (see 'Online' 'Write values' for setting only once at the beginning of a cycle) to user-defined values. The setting occurs in the run-time system, both at the beginning and at the end of the cycle.

The time sequence in one cycle: 1. Read inputs, 2. Force values 3. Process code, 4. Force values 5. Write outputs.

The function remains active until it is explicitly suspended by the user (command 'Online' 'Release force') or the programming system is logged-out.

For setting the new values, a **writelist** is first created, just as described under 'Online' 'Write values'. The variables contained in the writelist are accordingly marked in Monitoring. The writelist is transferred to a **forcelist** as soon as the command 'Online' 'Force values' is executed. It is possible that an active forcelist already exists, in which case it is updated as required. The writelist is then emptied and the new values displayed in red as 'forced'. Modifications of the forcelist are transferred to the program with the next 'Force values' command.

Note: The forcelist is created at the first forcing of the variables contained in the writelist, while the writelist existed prior to the first writing of the variables that it contains.

The command for forcing a variable, which means that it will be entered into the forcelist can be found at the following places:

- Command 'Force Values' in the menu 'Online'.
- Button 'Force Values' in the dialog 'Editing the writelist and the forcelist'.

Note:	In the sequential function chart language, the individual values from which a transition expression is assembled cannot be changed with 'Force values'. This is due to the fact that in monitoring the 'Total value' of the expression, not the values of the individual variables are displayed (e.g. "a AND b" is only displayed as TRUE if both variables actually have the value TRUE).
--------------	---

In FBD, on the other hand, only the first variable in an expression, used for example as input to a function block, is monitored. Thus a 'Force values' command is only possible for this variable.

'Online' 'Release force'**Shortcut: <Shift>+<F7>**

This command ends the forcing of variable values in the controller. The variable values change again in the normal way.

Forced variables can be recognized in Monitoring by the red color in which their values are displayed. You can delete the whole forcelist, but you can also mark single variables for which the forcing should be released.

To delete the whole forcelist, which means to release force **for all variables**, choose one of the following ways:

Command 'Release Force' in menu 'Online'.

Button 'Release Force' in dialog 'Editing the writelist and the forcelist'

Delete the whole forcelist using the command 'Release Force' in the dialog 'Remove Write-/Forcelist'. This dialog opens if you choose the command 'Release Force' while also a writelist exists.

To release force only **for single variables** you have to mark these variable first. Do this in one ways described in the following. After that the chosen variables are marked with an turquoise extension <Release Force>:

A double mouse click on a line, in which a non boolean variable is declared, opens the dialog 'Write variable <x>'. Press button <Release Force for this variable> .

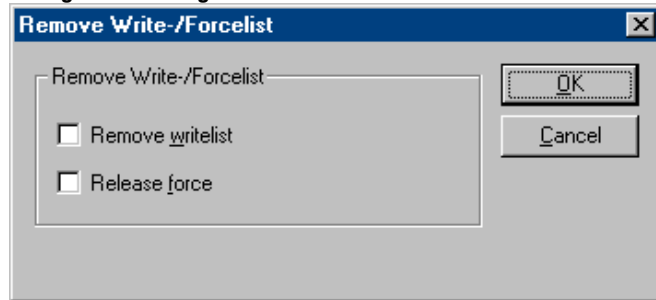
Repeat double mouse clicks on a line in which a boolean variable is declared to toggle to the display <Release Force> at the end of the line.

In the menu 'Online' open the Write/Force-Dialog and delete the value in the edit field of the column 'Forced value'.

When for all desired variables the setting "<Release Force>" is shown in the declaration window, choose the command 'Force values' to transfer the modifications of the forcelist to the program.

If the current writelist (see 'Online' 'Write Values') is not empty while you execute the command 'Release Force', the dialog 'Remove Write-/Forcelist' will be opened. There the user has to decide whether he just wants to **Release Force** or additionally wants to **remove the writelist** or if he wants to remove both lists.

Dialog for removing Write-/Forcelists

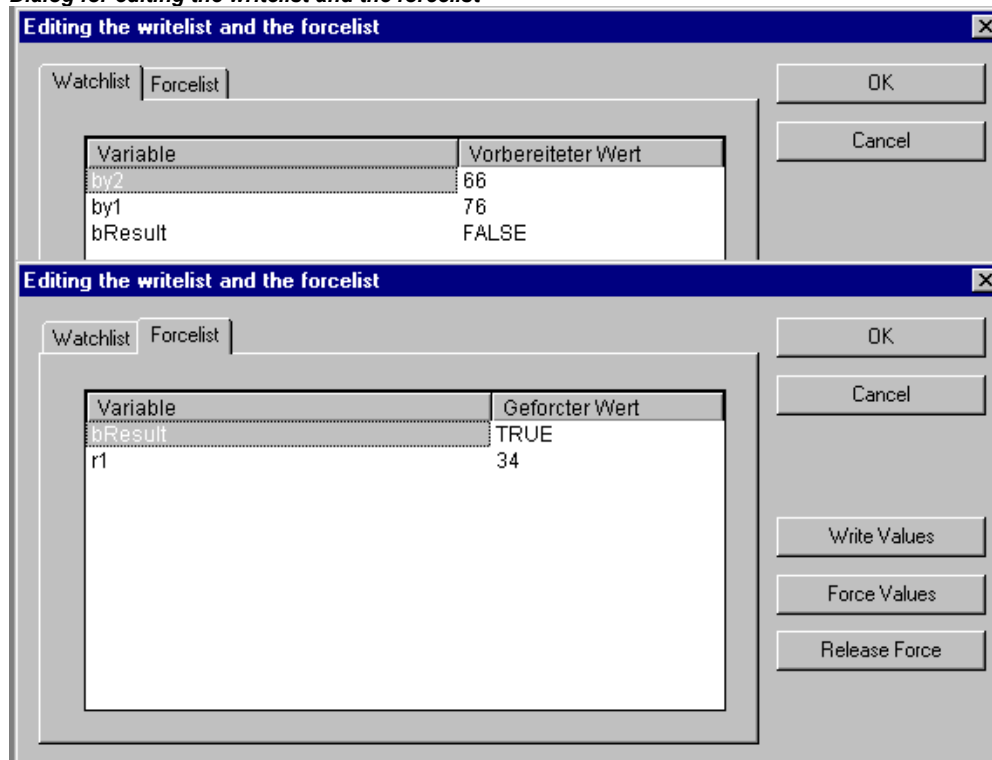


'Online' 'Write/Force' Dialog'

Shortcut: <Ctrl>+<Shift>+<F7>

This command leads to a dialog which displays in two registers the current writelist (**Watchlist**) and forcelist (**Forcelist**). Each variable name and the value to be written to it or forced on it are displayed in a table.

Dialog for editing the writelist and the forcelist



The variables reach the watchlist via the commands 'Online' 'Write Values' and are transferred to the forcelist by the command 'Online' 'Force Values'. The values can be edited here in the „Prepared Value" or „Forced Value" columns by clicking the mouse on an entry to open an editor field. If the entry is not type-consistent, an error message is displayed. If a value is deleted, it means that the entry is deleted from the writelist or the variable is noticed for suspension of forcing as soon as the dialog is closed with any other command than **Cancel**.

The following commands, corresponding to those in the Online menu, are available via buttons:

Force Values: All entries in the current writelist are transferred to the forcelist, that is the values of the variables in the controller are forced. All variables marked with 'Release Force' are no longer forced. The dialog is then closed.

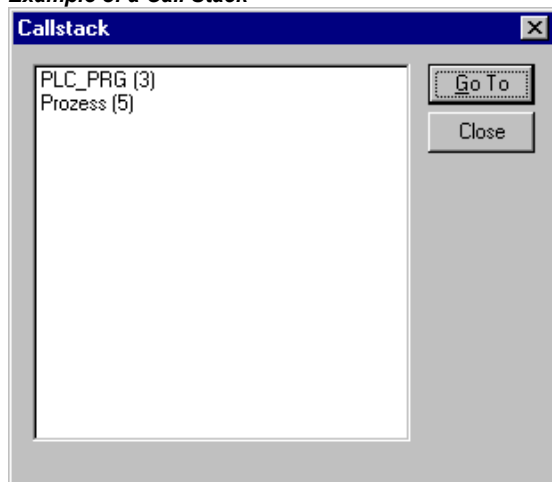
Write Values: All entries in the current writelist are written once only to the corresponding variables in the controller. The dialog is then closed.

Release Force: All entries in the forcelist will be deleted or, if a writelist is present, the dialog "Delete write-/forcelist" comes up, in which the user must decide whether he only wants to **release forcing** or **discard the writelist**, or both. The dialog will close at that point, or after the selection dialog is closed as the case may be.

'Online' 'Show Call Stack'

You can run this command when the Simulation Mode stops at a breakpoint. You will be given a dialog box with a list of the POU Call Stack.

Example of a Call Stack



The first POU is always PLC_PRG, because this is where the executing begins.

The last POU is always the POU being executed.

After you have selected a POU and have pressed the **Go to** button, the selected POU is loaded in its editor, and it will display the line or network being processed.

'Online' 'Display Flow Control'

Depending on the target system settings the user can activate resp. deactivate the Flow Control function. If it is activated, a check(✓) will appear in front of the menu item. Following this, every line or every network will be marked which was executed in the last PLC Cycle.

The line number field or the network number field of the lines or networks which just run will be displayed in green. An additional field is added in the IL-Editor in which the present contents of the accumulator are displayed. In the graphic editors for the Function Block Diagram and Ladder Diagram, an additional field will be inserted in all connecting lines not transporting any Boolean values. When these Out- and Inputs are verified, then the value that is transported over the connecting line will be shown in this field. Connecting lines that transport only Boolean values will be shaded blue when they transport TRUE. This enables constant monitoring of the information flow.

Please regard:

1. The runtime of a program will be increased by using flow control. This might cause timeouts in time-cyclic programs with high load.
2. At active breakpoint positions there is no flow control display.
3. If a watchdog has been defined for the concerned tasks (see Chapter 6.7.2, Insert task), this will be switched off when the flow control is active.

'Online' 'Simulation'

If **Simulation Mode** is chosen, then a check(✓) will appear in front of the menu item.

In the simulation mode, the user program runs on the same PC under Windows. This mode is used to test the project. The communication between the PC and Simulation Mode uses the Windows Message mechanism.

If the program is not in simulation mode, then the program will run on the PLC. The communication between the PC and the PLC typically runs over the serial interface.

The status of this flag is stored with the project.

Please regard:

- POU's of external libraries will not run in simulation mode.

'Online' 'Communication Parameters'

You are offered a special dialog for setting communication parameters when the communication between the local PC and the run-time system is running over a gateway server in your system. (If the OPC or DDE server is used, the same communication parameters must be entered in its configuration).

See the following items:

Principle of a gateway system

Communication Parameters Dialog for the local PC

Setting up the desired gateway server and channel

Setting up a new channel for the local gateway server

What the communication parameters dialog on the local PC shows

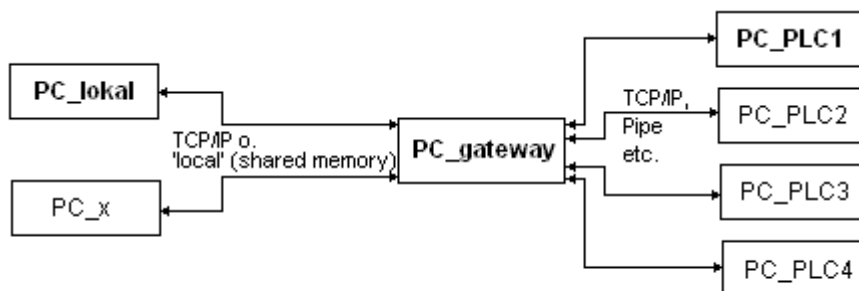
Principle of a gateway system

Principle of a gateway system

Let us examine the principle of the gateway system before explaining the operation of the dialog:

A gateway server can be used to allow your local PC to communicate with one or more run-time systems. The setting concerning which run-time systems can be addressed, which is specifically configured for each gateway server, and the connection to the desired gateway server, is made on the local PC. Here it is possible that both the gateway server and the run-time system(s) can run together on the local PC. If we are dealing with a gateway server which is running on another PC we must ensure that it has been started there. If you are selecting a locally installed gateway server, it automatically starts when you log onto the target run-time system. You can recognise this through the appearance of a **CoDeSys** symbol on the bottom right in the task bar. This symbol lights up as long as you are connected to the run-time system over the gateway. The menu points **Info** and **Finish** are obtained by clicking with the right mouse key on the symbol. **Finish** is used to switch off the gateway.

See the following scheme for presenting a gateway system:



PC_local is your local PC, **PC_x** is another PC, which gateway addresses. **PC_gateway** is the PC on which the gateway server is installed, **PC_PLC1** through to **PC_PLC4** are PCs on which the run-time systems are running. The diagram shows the modules as separated but it is fully possible for the Gateway server and / or run-time systems to be installed together on the local PC.

Important: Please note that a connection to gateway is only possible over TCP/IP so make sure that your PC is configured appropriately!

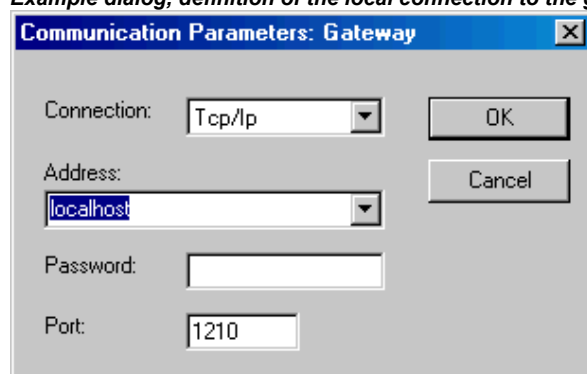
The connections from gateway to the various run-time computers can, on the other hand, run over different protocols (TCP/IP, Pipe, etc.).

Setting up the desired gateway server and channel

1. Setting up the desired gateway server and channel in the Communication Parameters dialog:

To define the connection to the desired gateway server we open the dialog 'Communication Parameters Gateway' by pressing the button **Gateway**.

Example dialog, definition of the local connection to the gateway



Here you can enter and/or edit the following:

The **type** of connection from your computer to the computer on which the gateway server that you want to use is running. If the gateway server is running on the local computer, connection via shared memory („local“) or via TCP/IP is possible; if connection to a different computer is needed, only TCP/IP can be used.

The **address** of the computer, on which the gateway server that you want to use is running: IP address or the appropriate symbolic name such as e.g. localhost. Regard that leading zeros added to the address range number however are not allowed (example: not possible: '010.107.084.050', must be entered as '10.107.84.50'). On initial setup, the standard 'localhost' is offered as the computer name (address), which means that the locally installed gateway would be accessed. The name 'localhost' is set to be identical to the local IP address 127.0.0.1 in most cases, but you may in some cases have to enter this directly into the Address field. If you want to access a gateway server on another computer, you must replace 'localhost' with its name or IP address.

The **password** for the selected gateway server, if it is on a remote computer. If it is incorrectly entered, or not entered at all, an error message appears.

Note in this connection: you can give the locally installed gateway server a password with the following procedure: click with the right mouse button on the gateway symbol in the lower right portion of the toolbar and select „Change password“. A dialog comes up for changing or entering a password. If you access the gateway server locally any password that is entered will not be asked for.

The computer's **port** on which the gateway server that you wish to use is running, as a rule the correct value for the selected gateway is already given.

If the dialog is closed with **OK**, the corresponding entry (computer address) appears in the **Channels** field at the top of the 'Communication parameters' dialog, and below it the channels available on this gateway server.

2. Setting up the desired channel on the selected gateway server:

Now select one of the channels by clicking on an entry with the mouse. The corresponding parameters will then be shown in the table. If no connection can be established to the selected gateway address — possibly because it has not been started or the address is incorrect — the phrase 'not connected' appears in brackets after the address and a message 'No gateway with these settings could be found' appears. In this connection perform a quick check.

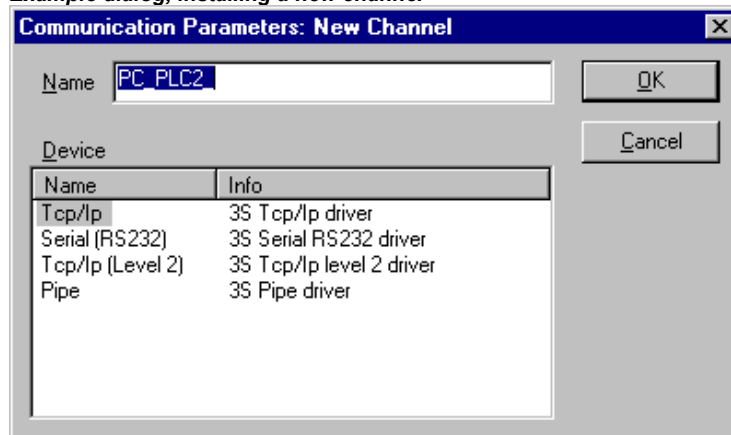
Once the desired channel is set up, close the dialog using **OK**. The settings are saved with the project.

Setting up a new channel for the local gateway server

You can set up new channels for the currently connected gateway server, which are then available for establishing further connection from the server, a connection to a controller for example. The options that you have in this regard depend on the particular choice of the number of device drivers installed on your computer.

Press the **New** button in the Communication Parameters dialog. The dialog **Communication Parameters: New Channel** comes up:

Example dialog, installing a new channel



The input field **Name** automatically contains the name used for the last inputted channel. If no channel has yet been defined, the current gateway name will be offered, followed by an underline character, e.g. 'localhost_'. You can edit the channel name at this point. The channel name is purely informative, it does not have to be a unique name but it is recommended to use one.

The **device** drivers available on the gateway computer are listed in the table under Device. In the Name column, select by mouse click one of the available drivers; the corresponding comment, if any, appears in the Info column.

If you close the '...New Channel' dialog with **OK**, the newly defined channel appears in the 'Communication Parameters' dialog as a new entry in **Channels** at the lowest position under the minus sign. So far, it is only stored locally in the project (see above). At this point you can edit the Value column (see tips below). Now confirm the entered parameters with OK, thus leaving the 'Communication Parameters' dialog.

In order for the newly entered gateway channel and its parameters to also be known to the gateway server xy, and thus also to make it available to other computers that access this gateway xy, you must **log into the run-time system**. If you then re-open the 'Online' 'Communication parameters' dialog, the new channel appears in the „channel tree“, not only in its previous position but also indented under the address or name of the gateway server xy. This indicates that it is known to the network. You can now open the Communication Parameter dialog on a computer other than the local one, select gateway xy and use its new channel.

If a communications error occurs when logging in, it is possible that the interface cannot be opened (e.g. COM1 for a serial connection) possibly because it is being used by another device. It is also possible that the controller is not running.

The parameters for a channel already known by the gateway server can no longer be edited in the configuration dialog. The parameter fields appear grey. You can, however, delete the connection as long as it is not active.

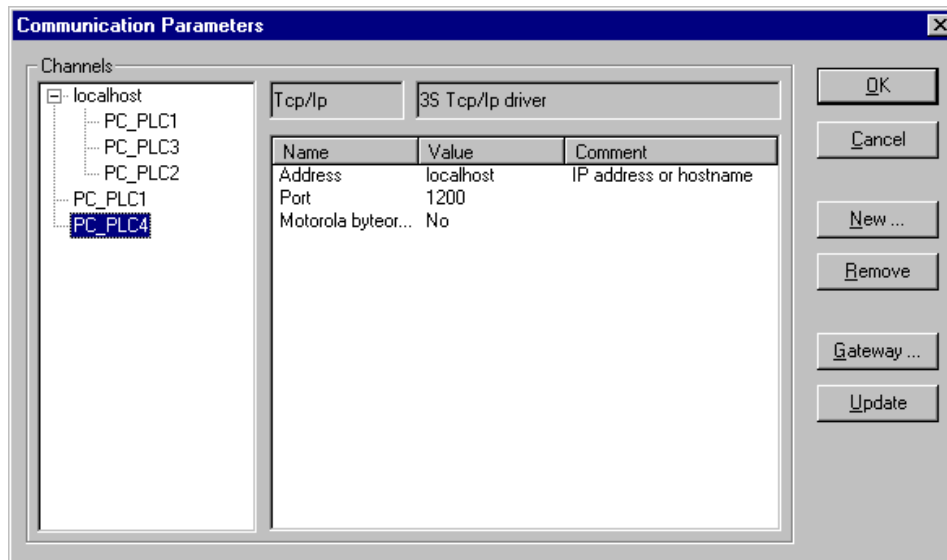
Important: Please note that the deletion of a channel is not reversible. It occurs at the moment that you press on the button **Remove** !

What the communication parameters dialog on the local PC shows

This dialog is used to select a gateway server for the communication with a PLC. Further on there can be set up new channels for a gateway server which is installed on the local PC so that these channels can be used by other computers which are part of the network.

The current settings can be called up at any time using the button **Update**.

The dialog will appear as follows if the communication parameters have already been configured according to the example in 'Principle of a gateway system':



The Heading **Channels** lists two categories of connections:

On the one hand all of the connections are shown which are installed on the currently connected gateway server called 'localhost'. Here the address or the name of this gateway is located on the upper position behind the minus sign, which in our example is running on the local computer. The appropriate address 'localhost' corresponds in the normal case to the IP address 127.0.0.1 of the local computer (PC_local). Below, indented to the right, are three addresses of run-time computers which the gateway channels are set-up to (PC_PLC1 to 3). They could have been configured both from the local PC or from the other PCs (PC_x) which are or were connected to the gateway server.

The second category of the channels describes includes all connections to the gateway which can be set up from your local PC, over this configuration dialog for example. They create the "branch" which leads from the minus sign directly below to PC_PLC1 and PC_PLC4. These channel addresses do not necessarily have to be known yet at the gateway. For PC_PLC4 in the example described above, the configuration parameters are stored locally in the project but they will first be known to the gateway the next time log-in to the run-time system occurs. This has already occurred for PC_PLC1 since the associated gateway address has appeared as an additional "sub-branch" to the "channel tree".

In the central part of the dialog one finds the designation, in each case, of the left selected channel and the associated parameter under **Name**, **Value** and **Comment**.

Tips for editing the parameters in the communication parameters dialogue

You can only edit the text fields in the column **Value**.

Select a text field with the mouse, and get into the editing mode by double clicking or by pressing the space bar. The text input is finished by pressing the <Enter> key.

You can use <Tabulator> or <Shift> + <Tabulator> to jump to the next or the previous switching or editing possibility.

To edit numerical values it is possible with the arrow keys or the Page Up/Down keys to change the value by one or ten units respectively. A double click with the mouse also changes the value by increasing by one unit. A typing check is installed for numerical values: <Ctrl> + <Home> or <Ctrl> + <End> deliver the lowest or the highest value respectively for the possible input values for the type of parameter in question.

Quick check in the event of unsuccessful connection attempt to the gateway

You should make the following checks if the connection to the selected gateway computer is not successful. (You get the message „not connected" in the Communication Parameters dialog behind the gateway server address in the field Channels):

Has the gateway server been started (the three-color symbol appears in the bottom right portion of the toolbar) ?

Is the IP address that you entered in the 'Gateway: Communication Parameters' dialog really that of the computer on which the gateway is running? (use „ping" to check)

Is the TCP/IP connection working locally? The error may possibly lie with TCP/IP.

'Online' 'Sourcecode download'

This command loads the source code for the project into the controller system. This is not to be confused with the Code that is created when the project is compiled! You can enter the options that apply to Download (time, size) in the 'Project' 'Options' 'Sourcedownload' dialog.

'Online' 'Create boot project'

With this command, the compiled project is set up on the controller in such a way that the controller can load it automatically when restarted. Storage of the boot project occurs differently depending on the target system. For example, on 386 systems three files are created: **default.prg** contains the project code, **default.chk** contains the code's checksum, **default.sts** contains the controller status after restart (start/stop).

The command 'Online' 'Create boot project' is also available in **offline mode** if the project has been built without errors. In this case the following files are created in the projects directory: **<projektname>.prg** for the boot project code, and **<projektname>.chk** for the checksum. These files can be renamed as necessary and then be copied to a PLC.

Depending on the target system settings at the creation of a boot project in offline mode a new *.ri-file (download information) might be created. Also depending on the target setting a dialog will appear if this file already exists.

Note: If the project option **Implicit at create boot project** (category Source download) is activated, then the selected sources will be loaded automatically into the controller on the command 'Online' 'Create boot project'.

'Online' 'Write file to controller'

This command is used for loading any desired file onto the controller. It opens the dialog for 'Write file to controller' in which you can select the desired file.

After the dialog is closed using the 'Open' button, the file is loaded into the controller and stored there under the same name. The loading process is accompanied by a progress dialog.

With the command 'Online' 'Load file from controller' you can retrieve a file previously loaded on the controller.

'Online' 'Load file from controller'

With this command, you can retrieve a file previously loaded into the controller using 'Online' 'Write file to controller'. You receive the 'Load file from controller' dialog. Under Filename, provide the name of the desired file, and in the selection window enter the directory on your computer into which it is to be loaded as soon as the dialog is closed with the „Save" button.

4.7 Window set up...

Under the **'Window'** menu item you will find all commands for managing the windows. There are commands for the automatic set up of your window as well as for opening the library manager and for changing between open windows. At the end of the menu you will find a list of all open windows in the sequence they were opened. You can switch to the desired window by clicking the mouse on the relevant entry. A check will appear in front of the active window.

'Window' 'Tile Horizontal'

With this command you can arrange all the windows horizontally in the work area so that they do not overlap and will fill the entire work area.

'Window' 'Tile Vertical'

With this command you can arrange all the windows vertically in the work area so that they do not overlap and will fill the entire work area.

'Window' 'Cascade'

With this command you can arrange all the windows in the work area in a cascading fashion, one behind another.

'Window' 'Arrange Symbols'

With this command you can arrange all of the minimized windows in the work area in a row at the lower end of the work area.

'Window' 'Close All'

With this command you can close all open windows in the work area.

'Window' 'Messages'

Shortcut: <Shift>+<Esc>

With this command you can open or close the message window with the messages from the last compiling, checking, or comparing procedure.

If the messages window is open, then a check (✓) will appear in front of the command.

4.8 Help when you need it...

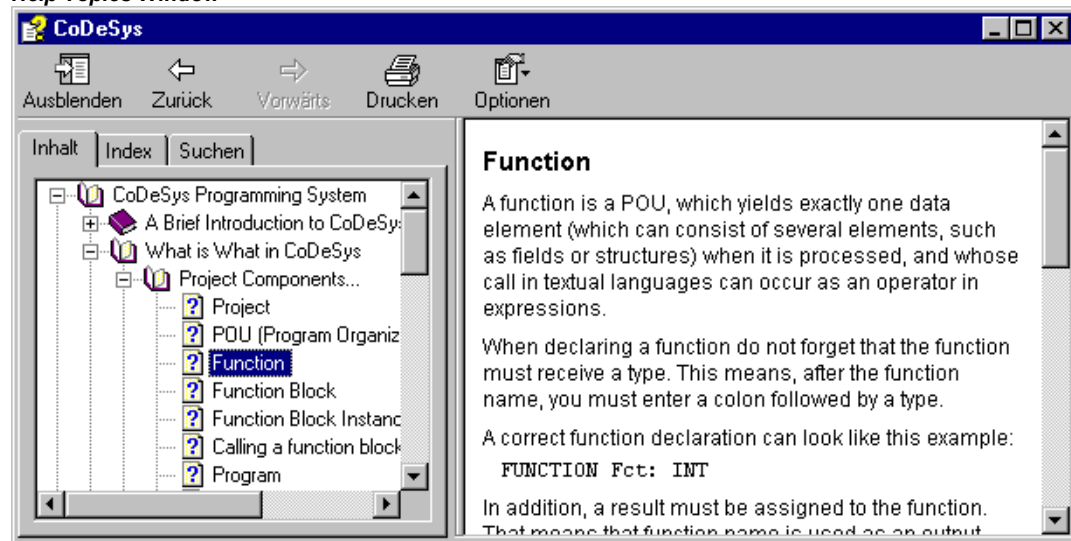
'Help' 'Contents and Search'

With the commands Contents resp. Search in the Help menu you can open the help topics window, which will be displayed via the HTML Help Viewer (Internet Explorer V4.1 and higher).

The **Contents** tab shows the contents tree. The books can be opened and closed by a double-click or via the plus and minus signs. That page which is currently selected in the contents tree will be displayed in the right part of the window. Hyperlinks from the text to other help pages resp. expanding hotspots are marked by a different color and an underline. A mouse-click on such texts will open the linked page resp. will show the expanded text or a picture.

In the **Index** tab you can look for help pages on specific items, in the **Search** tab a full-text search on all help pages can be done. Follow the instructions in the register cards.

Help Topics Window



Context Sensitive Help

Shortcut: <F1>

You can use the <F1> key in an active window, in a dialog box, or above a menu command in order to open the online help. When you perform a command from the menu, the help for the command called up at that time is displayed.

You can also highlight a text (for example, a key word or a standard function) and press <F1> to have the help displayed for that item.

Help when you need it...

5 Editors in CoDeSys

5.1 This is for all Editors...

Components of an Editor

All editors for POUs (Program Organization Units) consist of a declaration part and a body. The body can consist of other a text or a graphic editor; the declaration portion is always a text editor. Body and declaration part are separated by a screen divider that can be dragged, as required, by clicking it with the mouse and moving it up or down.

Print margins

The vertical and horizontal margins that apply when the editor contents are printed, are shown by red dashed lines if the **'Show print range'** option in the project options in the dialog **'Workspace'** was selected. The properties of the printer that was entered apply, as well as the size of the print layout selected in the 'File' 'Printer Setup' menu. If no printer setup or no print layout is entered, a default configuration is used (Default.DFR and default printer). The horizontal margins are drawn as if the options 'New page for each object' or 'New page for each sub-object' were selected in 'Documentation settings'. The lowest margin is not displayed.

Note: An exact display of the print margins is only possible when a zoom factor of 100% is selected.

Comment

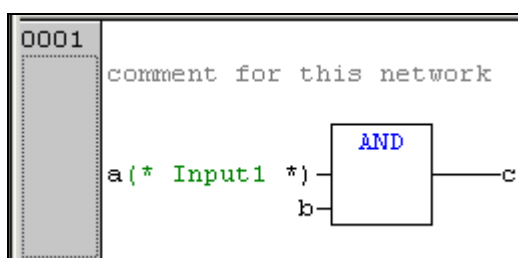
User comments must be enclosed in the special symbol sequences „(*)“ and „*)“. Example: (*This is a comment.*)

Comments are allowed in all text editors, at any location desired, that is in all declarations, in the IL and ST languages and in self-defined data types. If the Project is printed out using a **template**, the comment that was entered during variable declaration appears in text-based program components after each variable.

In the FBD and LD graphic editors, comments can be entered for each network. To do this, search for the network on which you wish to comment and activate **'Insert' 'Comment'**.

Besides that comments always can be added where variable names are inserted.

Example in FBD for a network comment and for a comment placed behind an input variable:



In KOP a comment also can be added to each contact resp. each coil, if this is configured accordingly in the display options in menu 'Extras' 'Options'. In the Ladder Editor additionally a comment for each particular contact and coil can be added, if the corresponding options are activated in the menu 'Extras' 'Options'.

In CFC there are special comment POUs which can be placed at will.

In SFC you can enter comments about a step in the dialog for editing step attributes.

Nested comments are also allowed if the appropriate option in the 'Project' 'Options' 'Build Options' dialog is activated.

In Online mode, if you rest the mouse cursor for a short time on a variable, the type and if applicable the address and comment of that variable are displayed in a **tooltip**.

Zoom to POU

Shortcut: <Alt>+<Enter>

With this command a selected POU is loaded into its editor. The command is available in the context menu (<F2>) or in the 'Extras' menu, if the cursor is positioned on the name of a POU in a text editor or if the POU box is selected in a graphic editor.

If you are dealing with a POU from a library, then the library manager is called up, and the corresponding POU is displayed.

Open instance

This command corresponds to the 'Project' 'Open instance' command.

It is available in the context menu (<F2>) or in the 'Extras' menu, if the cursor is positioned on the name of a function block in a text editor or if the function block box is selected in a graphic editor.

Enter variables via the "Intellisense Function"

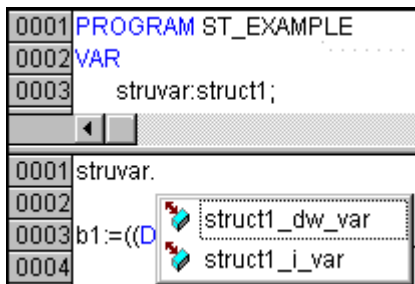
If the option List components is activated in the project options dialog for category 'Editor' , then the "Intellisense" functionality will be available in all editors, in the Watch- and Receiptmanager, in the Visualization and in the Sampling Trace:

If you insert a dot "." instead of an identifier, a selection box will appear, listing all local and global variables of the project. You can choose one of these elements and press 'Return' to insert it behind the dot. You can also insert the element by a double-click on the list entry.

If you enter a function block instance or a structure variable followed by a dot, then a selection box listing all input and output variables of the corresponding function block resp. listing the structure components will appear, where you can choose the desired element and enter it by pressing 'Return' or by a double-click.

Example:

Insert "struvar." -> the components of structure struct1 will be offered:



If you enter any string and press <Ctrl> + <Space Bar>, a selection box will appear listing all POUs and global variables available in the project. The list entry starting with the given string will be selected and can be entered to the program by pressing the <Enter> key.

Offline tooltip for Identifiers

In offline mode in all editors the following applies: If the cursor is placed on an editable identifier, in a tooltip the identifier's name, the variable class (e.g. VAR_GLOBAL), the data type, the variable's attributes (e.g. RETAIN), address and comment will be displayed.

5.2 Declaration Editor

5.2.1 Working in the Declaration Editor

The declaration editor is used to declare variables of POU and global variables, for data type declarations, and in the Watch and Receipt Manager. It gives access to the usual Windows functions, and even those of the IntelliMouse can be used if the corresponding driver is installed.

In Overwrite mode, **'OV'** is shown in black on the status bar; switching between Overwrite and Insert modes can be accomplished with the <Ins> key.

The declaration of variables is supported by syntax coloring.

The most important commands are found in the context menu (right mouse button or <Ctrl>+<F10>).

Hint: Regard the possibility of using pragmas to affect the properties of a variable concerning the compilation resp. precompilation process (see chapter 5.2.3).

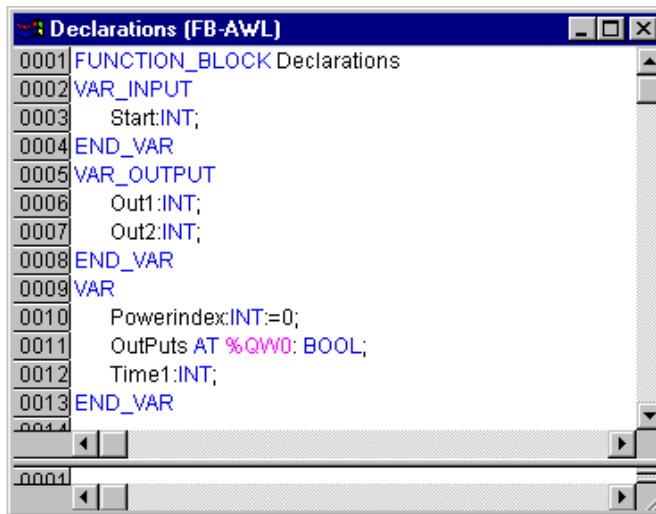
Declaration Part

All variables to be used only in this POU are declared in the declaration part of the POU. These can include: input variables, output variables, input/output variables, local variables, remanent variables and constants. The declaration syntax is based on the IEC61131-3 standard.

For a recommendation on the use of identifiers please see Appendix J: .

Regard the possibility of using templates for objects of type 'Global Variables', 'Data types', 'Function', 'Function Block' or 'Program' , see Chapter 4.3, 'File' New from template'.

An example of a correct declaration of variables in **CoDeSys**-Editor:



Input Variable

Between the key words **VAR_INPUT** and **END_VAR**, all variables are declared that serve as input variables for a POU. That means that at the call position, the value of the variables can be given along with a call.

Example:

```

VAR_INPUT
    iIn1:INT (* 1. Inputvariable*)
END_VAR

```

Output Variable

Between the key words **VAR_OUTPUT** and **END_VAR**, all variables are declared that serve as output variables of a POU. That means that these values are carried back to the POU making the call. There they can be answered and used further.

Example:

```
VAR_OUTPUT
  iOut1:INT; (* 1. Outputvariable*)
END_VAR
```

Input and Output Variables

Between the key words **VAR_IN_OUT** and **END_VAR**, all variables are declared that serve as input and output variables for a POU.

Attention: With this variable, the value of the transferred variable is changed ("transferred as a pointer", Call-by-Reference). That means that the input value for such variables cannot be a constant. For this reason, even the VAR_IN_OUT variables of a function block can not be read or written directly from outside via <functionblockinstance><in/outputvariable>.

Example:

```
VAR_IN_OUT
  iInOut1:INT; (* 1. Inputoutputvariable *)
END_VAR
```

Local Variables

Between the keywords **VAR** and **END_VAR**, all of the local variables of a POU are declared. These have no external connection; in other words, they can not be written from the outside.

Example:

```
VAR
  iLoc1:INT; (* 1. Local Variable*)
END_VAR
```

Remanent variables

Remanent variables can retain their value throughout the usual program run period. These include Retain variables and Persistent variables.

Example:

```
VAR RETAIN
  iRem1:INT; (* 1. Retain variable*)
END_VAR
```

Retain variables are identified by the keyword **RETAIN**. These variables maintain their value even after an uncontrolled shutdown of the controller as well as after a normal switch off and on of the controller (resp. at the command 'Online' 'Reset', see Chapter 4.6). When the program is run again, the stored values will be processed further. A concrete example would be an piece-counter in a production line, that recommences counting after a power failure.

All other variables are newly initialized, either with their initialized values or with the standard initializations.

Contrary to Persistent variables Retain Variables are reinitialized at a new download of the program.

Persistent variables are identified by the keyword **PERSISTENT**. Unlike Retain variables, these variables retain their value only after a re-Download, but not after an 'Online' 'Reset', 'Online' 'Reset (original)' or 'Online' 'Reset (cold)' (see Chapter 4.6 each), because they are not saved in the "retain area". If also persistent variables should maintain their values after a uncontrolled shutdown of the controller, then they have to be declared additionally as VAR RETAIN variables. A

concrete example of "persistent Retain-Variables" would be a operations timer that recommences timing after a power failure.

Attention:

- If a local variable in a **program** is declared as VAR RETAIN, then exactly that variable will be saved in the retain area (like a global retain variable)
- If a local variable in a **function block** is declared as VAR RETAIN, then the complete instance of the function block will be saved in the retain area (all data of the POU), whereby only the declared retain variable will be handled as a retain.
- If a local variable in a **function** is declared as VAR RETAIN, then this will be without any effect. The variable will **not** be saved in the retain area! If a local variable is declared as PERSISTENT in a function, then this will be without any effect also!

x = value will be retained - = value gets reinitialized

after Online command	VAR	VAR RETAIN	VAR PERSISTENT	VAR RETAIN PERSISTENT VAR PERSISTENT RETAIN
Reset	-	x	-	x
Reset (Kalt)	-	-	-	-
Reset (Ursprung)	-	-	-	-
Download	-	-	x	x
Online Change	x	x	x	x

Constants, Typed Literals

Constants are identified by the key word **CONSTANT**. They can be declared locally or globally.

Syntax:

```
VAR CONSTANT
<Identifier>:<Type> := <initialization>;
END_VAR
```

Example:

```
VAR CONSTANT
  c_iCon1:INT:=12; (* 1. Constant*)
END_VAR
```

See Appendix B: Operands in CoDeSys, for a listing of possible constants. See there also regarding the possibility of using typed constants (Typed Literals).

External variables

Global variables which are to be imported into the POU are designated with the keyword **EXTERNAL**. They also appear in the Watch window of the declaration part in Online mode.

If the VAR_EXTERNAL declaration does not match the global declaration in every respect, the following error message appears: "Declaration of '<var>' does not match global declaration!"

If the global variable does not exist, the following error message appears: "Unknown global variable: '<var>!'"

Example:

```
VAR EXTERNAL
  iVarExt1:INT:=12; (* 1st external variable *)
END_VAR
```

Keywords

Keywords are to be written in uppercase letters in all editors. Keywords may not be used as variables. Examples for keywords: VAR, VAR_CONSTANT, IF, NOT, INT.

Variables declaration

A variables declaration has the following syntax:

```
<Identifier> {AT <Address>};<Type> {:=<initialization>};
```

The parts in the braces {} are optional.

Regarding the identifier, that is the name of a variable, it should be noted that it may not contain spaces or umlaut characters, it may not be declared in duplicate and may not be identical to any keyword. Upper/lowercase writing of variables is ignored, in other words VAR1, Var1 and var1 are not different variables. Underlines in identifiers are meaningful, e.g. A_BCD and AB_CD are interpreted as different identifiers. Multiple consecutive underlines at the beginning of an identifier or within a identifier are not allowed. The length of the identifier, as well as the meaningful part of it, is unlimited.

All declarations of variables and data type elements can include initialization. They are brought about by the "!=" operator. For variables of elementary types, these initializations are constants. The default-initialization is 0 for all declarations.

Example:

```
iVar1:INT:=12; (* Integer variable with initial value of 12*)
```

If you wish to link a variable directly to a definite address, then you must declare the variable with the keyword **AT**.

For faster input of the declarations, use the shortcut mode.

In function blocks you can also specify variables with incomplete address statements. In order for such a variable to be used in a local instance, there must be an entry for it in the variable configuration.

Pay attention to the possibilities of an automatic declaration resp. of using pragmas to affect the properties of variables concerning the compilation process.

AT Declaration

If you wish to link a variable directly to a definite address, then you must declare the variable with the keyword **AT**. The advantage of such a procedure is that you can assign a meaningful name to an address, and that any necessary changes of an incoming or outgoing signal will only have to be made in one place (e.g., in the declaration).

Notice that variables requiring an input cannot be accessed by writing.

Examples:

```
xCounterHeat7 AT %QX0.0: BOOL;
wLightcabinetimpulse AT %IW2: WORD;
xDownload AT %MX2.2: BOOL;
```

Note:	If boolean variables are assigned to a Byte, Word or DWORD address, they occupy one byte with TRUE or FALSE, not just the first bit after the offset!
--------------	---

'Insert' 'Declaration keywords'

You can use this command to open a list of all the keywords that can be used in the declaration part of a POU. After a keyword has been chosen and the choice has been confirmed, the word will be inserted at the present cursor position.

You also receive the list, when you open the Input Assistant (<F2>) and choose the **Declarations** category.

'Insert' 'Type'

With this command you will receive a selection of the possible types for a declaration of variables. You also receive the list when you access the Input Assistant (<F2>).

The types are divided into the following categories:

Standard types BOOL, BYTE, etc.

Defined types Structures, enumeration types, etc.

Standard function blocks for instance declarations

Defined function blocks for instance declarations

CoDeSys supports all standard types of IEC1131-3:

Syntax Coloring

In all editors you receive visual support in the implementation and declaration of variables. Errors are avoided, or discovered more quickly, because the text is displayed in color.

A comment left unclosed, thus annotating instructions, will be noticed immediately; keywords will not be accidentally misspelled, etc.

The following color highlighting will be used:

Blue	Keywords
Green	Comments in the text editors
Pink	Special constants (e.g. TRUE/FALSE, T#3s, %IX0.0)
Red	Input error (for example, invalid time constant, keyword, written in lower case,...)
Black	Variables, constants, assignment operators, ...

Shortcut Mode

The declaration editor for **CoDeSys** allows you to use the shortcut mode. This mode is activated when you end a line with <Ctrl><Enter>

The following shortcuts are supported:

All identifiers up to the last identifier of a line will become declaration variable identifiers

The type of declaration is determined by the last identifier of the line. In this context, the following will apply:

B or BOOL	gives the result	BOOL
I or INT	gives the result	INT
R or REAL	gives the result	REAL
S or string	gives the result	STRING

If no type has been established through these rules, then the type is BOOL and the last identifier will not be used as a type (Example 1.).

Every constant, depending on the type of declaration, will turn into an initialization or a string (Examples 2. and 3.).

An address (as in %MD12) is extended around the ATATDeclaration>Proc... attribute(Example 4.).

A text after a semicolon (;) becomes a comment (Example 4.).

All other characters in the line are ignored (e.g., the exclamation point in Example 5.).

Examples:

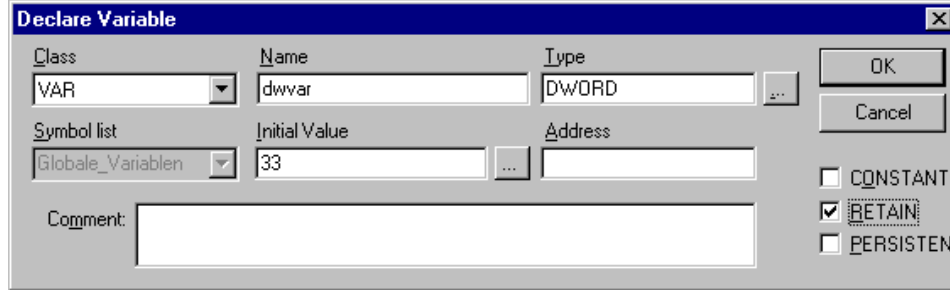
Shortcut	Declaration
A	A: BOOL;
A B I 2	A, B: INT := 2;
ST S 2; A string	ST:STRING(2); (* A string *)

X %MD12 R 5 Real Number X AT %MD12: REAL := 5.0;(* Real Number *)
 B ! B: BOOL;

Autodeclaration


If the Autodeclaration option has been chosen in the Editor category of the Options dialog box , then a dialog box will appear in all editors after the input of a variable that has not yet been declared. With the help of this dialog box, the variable can now be declared.

Dialog Box for Declaration of Variables



With the help of the **Class** combo box , select whether you are dealing with a local variable (**VAR**), input variable (**VAR_INPUT**), output variable (**VAR_OUTPUT**), input/output variable (**VAR_INOUT**), or a global variable (**VAR_GLOBAL**).

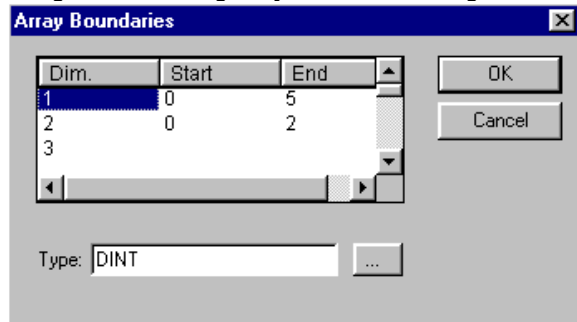
With the **CONSTANT**, **RETAIN**, **PERSISTENT** options, you can define whether you are dealing with a constant or a retain variable


The variable name you entered in the editor has been entered in the **Name** field, BOOL has been placed in the **Type** field. The  button opens the Input Assistant dialog which allows you to select from all possible data types.

Declaration of Arrays:


If ARRAY is chosen as the variable type, the dialog for entering array boundaries appears.

Dialog for determining array boundaries during automatic declaration



For each of the three possible dimensions (**Dim.**), array boundaries can be entered under **Start** and **End** by clicking with the mouse on the corresponding field to open an editing space. The array data type is entered in the **Type** field. In doing this, the  button can be used to call up an input assistant dialog.

Upon leaving the array boundaries dialog via the **OK** button, variable declarations in IEC format are set up based on the entries in the Type field in the dialog. Example: ARRAY [1..5, 1..3] OF INT

In the field **Initial value**, you may enter the initial value of the variable being declared. If this is an array or a valid structure, you can open a special initialization dialog via the  button or open the input assistant dialog for other variable types.

In the initialization dialog for an array you are presented a list of array elements; a mouse click on the space following „:=“opens an editing field for entering the initial value of an element.

In the initialization dialog for a structure, individual components are displayed in a tree structure. The type and default initial value appear in brackets after the variable name; each is followed by „:=“. A mouse click on the field following „:=“ opens an editing field in which you can enter the desired initial value. If the component is an array, then the display of individual fields in the array can be expanded by a mouse click on the plus sign before the array name and the fields can be edited with initial values.

After leaving the initialization dialog with **OK**, the initialization of the array or the structure appears in the field **Initial value** of the declaration dialog in IEC format.

Example: `x:=5,field:=2,3,struct2:=(a:=2,b:=3)`

In the **Address** field, you can bind the variable being declared to an IEC address (AT declaration).

If applicable, enter a **Comment**. The comment can be formatted with line breaks by using the key combination <Ctrl> + <Enter>.

By pressing **OK**, the declaration dialog is closed and the variable is entered in the corresponding declaration editor in accordance to the IEC syntax.

Note:	The dialog box for variable declaration you also get by the command 'Edit' 'Declare Variable'. If the cursor is resting on a variable in Online mode, the Autodeclare window can be opened with <Shift><F2> with the current variable-related settings displayed.
--------------	---

Line Numbers in the Declaration Editor

In offline mode, a simple click on a special line number will mark the entire text line.

In the online mode, a single click on a specific line number will open up or close the variable in this line, in case a structural variable is involved.

Declarations as tables

If the **Declarations as tables** option is activated in the Project Options (category 'Editor') or – if you already are working in the declaration editor – in the context menu, the declaration editor will look like a table. As in a card-index box, you can select the register cards of the respective variable types and edit the variables.

For each variable you are given the following entry fields.

Name: Input the identifier of the variable.

Address: If necessary, input the address of the variable (AT declaration)

Type: Input the type of the variable. (Input the function block when instantiating a function block)

Initial: Enter a possible initialization of the variable (corresponding to the " := " assignment operator).

Comment: Enter a comment here.

Both of the display types of the declaration editor can be changed without causing any problems. In online mode there are no different display types.

Insert new declaration:

In order to edit a new variable, select the 'Insert' 'New Declaration' command (see below).

Sorting the declarations:

In order to sort the table entries, set the cursor to the line number bar at the left border of the editor window and choose one of the following commands in the context menu:

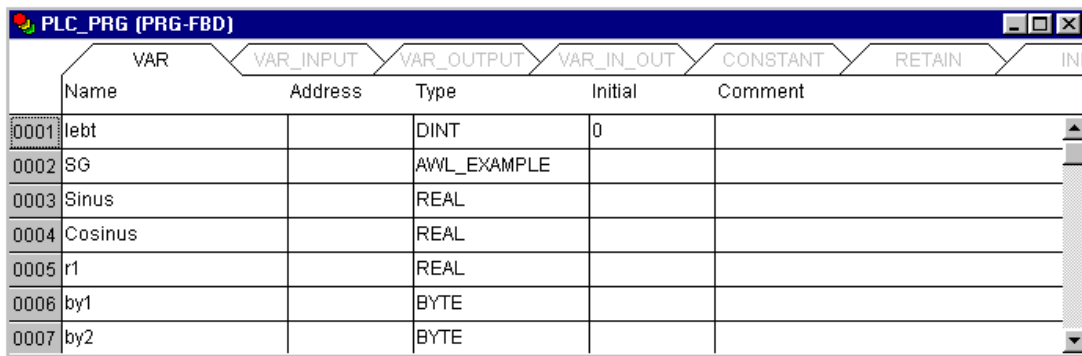
Sort by name: All lines are sorted alphabetically according to the identifier names in column 'Name'.

Sort by address: All lines are sorted alphabetically according to the address entries in column 'Address'.

Sort by type: All lines are sorted alphabetically according to the type names in column 'Type'.

Original order: The lines are displayed in the order in which they had been entered originally.

Declaration Editor as a Table



	VAR	VAR_INPUT	VAR_OUTPUT	VAR_IN_OUT	CONSTANT	RETAIN	INI
	Name	Address	Type	Initial	Comment		
0001	lebt		DINT	0			
0002	SG		AWL_EXAMPLE				
0003	Sinus		REAL				
0004	Cosinus		REAL				
0005	r1		REAL				
0006	by1		BYTE				
0007	by2		BYTE				

'Insert' 'New Declaration'

With this command you bring a new variable into the declaration table of the declaration editor. If the present cursor position is located in an field of the table, then the new variable will be pasted in the preceding line; otherwise, the new variable is pasted at the end of the table. Moreover, you can paste a new declaration at the end of the table by using the right arrow key or the tab key in the last field of the table.

You will receive a variable that has "Name" located in the **Name** field, and "Bool" located in the **Type** field, as its default setting. You should change these values to the desired values. Name and type are all that is necessary for a complete declaration of variables.

5.2.2 Declaration Editors in Online Mode

In online mode, the declaration editor changes into a monitor window. In each line there is a variable followed by the equal sign (=) and the value of the variable. If the variable at this point is undefined, three question marks (???) will appear. For function blocks, values are displayed only for open instances (command: 'Project' 'Open instance').

In front of every multi-element variable there is a plus sign. By pressing <Enter> or after doubleclicking on such a variable, the variable is opened up. In the example, the traffic signal structure would be opened up.

```

E---AMPEL1
  .....STATUS = 3
  .....GRUEN = FALSE
  .....GELB = FALSE
  .....ROT = TRUE
  .....AUS = FALSE
  
```

When a variable is open, all of its components are listed after it. A minus sign appears in front of the variable. If you double-click again or press <Enter>, the variable will be closed, and the plus sign will reappear.

Pressing <Enter> or double-clicking on a single-element variable will open the dialog box to write a variable. Here it is possible to change the present value of the variable. In the case of Boolean variables, no dialog box appears; these variables are toggled.

The new value is displayed after the variable, in pointed brackets and in turquoise color, and remains unchanged. If the 'Online' 'Write values' command is given, then all variables are placed in the selected list and are once again displayed in black.

If the 'Online' 'Force values' command is given, then all variables will be set to the selected values, until the 'Release force' command is given. In this event, the color of the force value changes to red

5.2.3 Pragma instructions in the Declaration Editor

The pragma instruction is used to affect the properties of a variable concerning the compilation resp. precompilation process. It can be used in with supplementary text in a program line of the declaration editor or in its own line.

The pragma instruction is enclosed in curly brackets, upper- and lower-case are ignored: { <Instruction text> }

If the compiler cannot meaningfully interpret the instruction text, the entire pragma is handled as a comment and read over. A warning is issued: „Ignore compiler directive ‚<Instruction text>’!”

Depending on the type and contents of pragma, the pragma either operates on the line in which it is located or on all subsequent lines until it is ended by an appropriate pragma, or the same pragma is executed with different parameters, or the end of the file is reached. By file we mean here: declaration part, implementation portion, global variable list, type declaration.

The opening bracket may immediately follow a variable name. Opening and closing brackets must be located on the same line.

The following pragmas are currently available in CoDeSys:

- Pragma {flag ...} for Initialization, Monitoring, Creation of symbols
- Pragma {bitaccess ...} for the Bitaccess
- Pragma {link} for Linking a POU or DUT
- Pragma {parameter ..}, {template...}, {instance...} for creating for Parameter Manager entries
- Pragma {library ...} for controlling the display of library declaration parts

Pragma instructions for Initialization, Monitoring, Creation of symbols, Bitaccess, Linking

Pragma {flag [<flags>] [off|on]}

<flags> can be a combination of the following flags:

noinit:	The variable will not be initialized.
nowatch:	The variable can not be monitored
noread:	The variable is exported to the symbol file without read permission
nowrite:	The variable is exported to the symbol file without write permission
noread, nowrite:	The variable will not get exported to the symbol file

With the „on” modifier, the pragma operates on all subsequent variable declarations until it is ended by the pragma {flag off}, or until overwritten by another {flag <flags> on} pragma.

Without the „on” or „off” modifier, the pragma operates only on the current variable declaration (that is the declaration that is closed by the next semicolon).

Examples for use of pragma {flag}:

Initialization and monitoring of variables:

The variable a will not be initialized and will not be monitored. The variable b will not be initialized:

```
VAR
  a : INT {flag noinit, nowatch};
  b : INT {flag noinit };
END_VAR

VAR
  {flag noinit, nowatch on}
  a : INT;
  {flag noinit on}
  b : INT;
  {flag off}
```

```
END_VAR
```

Neither variable will be initialized:

```
{flag noinit on}
VAR
  a : INT;
  b : INT;
END_VAR
{flag off}
VAR
{flag noinit on}
  a : INT;
  b : INT;
{flag off}
END_VAR
```

Getting variables to the symbol file:

The flags „**noread**“ and „**nowrite**“ are used, in a POU that has read and/or write permission, to provide selected variables with restricted access rights. The default for the variable is the same as the setting for the POU in which the variable is declared. If a variable has neither read nor write permission, it will not be exported into the symbol file.

Examples:

If the POU has read and write permission, then with the following pragmas variable a can only be exported with write permission, while variable b can not be exported at all:

```
VAR
  a : INT {flag noread};
  b : INT {flag noread, nowrite};
END_VAR
VAR
{ flag noread on}
  a : INT;
{ flag noread, nowrite on}
  b : INT;
{flag off}
END_VAR
```

Neither variable a nor b will be exported to the symbol file:

```
{ flag noread, nowrite on }
VAR
  a : INT;
  b : INT;
END VAR
{flāg off}
VAR
{ flag noread, nowrite on}
  a : INT;
  b : INT;
{flag off}
END_VAR
```

The pragma operates additively on all subsequent variable declarations.

Example: (all POUs in use will be exported with read and write permission)

```
a : afb;
...
FUNCTION_BLOCK afB
VAR
  b : bfb {flag nowrite};
  c : INT;
END_VAR
...
FUNCTION_BLOCK bfB
VAR
  d : INT {flag noread};
```

```
e : INT {flag nowrite};
END_VAR
```

„a.b.d“: Will not be exported

„a.b.e“: Will be exported only with read permission

„a.c“: Will be exported with read and write permission.

Pragma {bitaccess...} for the Bitaccess

This pragma can be used to get a correct display of a variable, which is doing a bitaccess with the help of a global constant, in the input assistant, in the "Intellisense function" and at monitoring in the declaration window. Further on it will effect that, when this variable is monitored in the declaration window of the particular POU, the used global constants are shown below the respective structure variable.

Please regard: The project option 'Replace constants' (category Build) must be activated !

The pragma must be inserted in the declaration of the structure in a separate line. The line is not terminated by a semicolon.

Syntax:

```
{bitaccess <Global Constant> <Bitnumber> '<comment>'}
```

<Global Constant>: Name of the global constant, which must be defined in a global variables list.

<Bitnumber>: Value of the global constant, as defined in the global variables list.

See for an example in Chapter Appendix B: Operands in CoDeSys, Addressing bits in variables.

Pragma {link} for linking a POU during code generation

Normally a POU (program, function, function block) or a data unit type definition (DUT) which is not called within the project, will not be linked during code generation. But it might be desired that a function, e.g. included in the project via a library, should be available after download on the runtime system even if it is not used by the application program directly (e.g. for any check operations). For this purpose you can add the {link} pragma at any desired position in the declaration part of a POU or in a DUT in order to force a linking of the POU anyway.

Pragma for Controlling the Display of Library Declaration Parts

During creation of a library in CoDeSys you can define via pragmas which parts of the declaration window should be visible resp. not visible in the Library Manager later when the library will be included in a project. The display of the implementation part of the library will not be affected by that.

Thus comments or any variables declarations can be concealed from the user. The pragmas {library private} and {library public} each affect the rest of the same line resp. the subsequent lines, as long as they are not overwritten by the each other one.

Syntax: {library public} The subsequent test will be displayed in the Library Manager.

{library private} :The subsequent test will be not displayed.

Example: See below the declaration part of a library, which is created in CoDeSys. The comment "(* this is for all *)" should be displayed in the Library Manager after having included the library in a project., the comment "(* but this is not for all *)" however should not be displayed. The variables local and in2 also should not be displayed:

```
{library public}(*this is for all*) {library private} (*this is not for all*)
{library public}
FUNCTION afun : BOOL
```

```

VAR_INPUT
  in: BOOL;
END_VAR

(library private)

VAR
  local: BOOL;
END_VAR

(library public)

VAR_INPUT
  in2: BOOL;
  {library private}
  in3: BOOL;
  {library public}
END_VAR

```

Pragma instructions for Parameter Manager entries

Pragma instructions can be inserted in variable declarations in order to create entries for these variables in parameter lists which are handled in the Parameter Manager. It depends on the target system, whether the Parameter Manager is available in the CoDeSys programming system. This means it must be activated in the target settings, category Networkfunctionality.

The syntax:

The pragma instruction is enclosed in curly brackets, upper- and lower-case are ignored: { <Instruction text> }. If it is included in "normal" variable declarations, it must be set before the terminating semicolon of the declaration.

Pragmas, which are used in the VAR_CONFIG interface, are set each in a single line and are not terminated with a semicolon !

The key definitions are written separated by space characters, all together enclosed in square brackets.

<name>: Name of the parameter list in the Parameter Manager. If the list does not yet exist, it will be created.

<key>: Name of the attribute, i.e. column title in the parameter list; e.g. "Name", "Value", "Accesslevel" etc.; It depends on the definition of the customer specific parameter list type, which keys can be defined in the pragma. The key definitions are written separated by space characters, all together enclosed in square brackets. Regard the syntax for entries in instance lists for arrays, structure resp. function block components.(see below, 3.).

<value>: Value of the attribute which is defined by <key>. If <value> contains empty spaces it must be enclosed in double quotation marks, e.g. ...accessright="read only"...

Please note: The pragma instructions will be effective as soon as the focus is changed, that means as soon as a precompile is done e.g. via leaving the editor window. Erroneous inputs will not be messaged until the project is compiled.

The following entries can be generated:

1. Entries in parameter lists of type 'Variables'

(a) from the declaration part of programs and global variables lists:

For a variable defined in a PROGRAM- or VAR_GLOBAL declaration an entry in a parameter list of type 'Variables' can be created, if it is declared like described in the following: (If the parameter list does not yet exist, it will be created automatically)

Syntax: {parameter list=<name> [<key>=<value> <key>=<value> ...further keys] }

Example: Variable bvar is declared in a program. It should be entered in the parameter list parlist1 type 'Variables') with symbolic name bvar1, value 102, index 16#1200 and subindex 16#21.

```
VAR
```

```
bvar:INT{parameter list=parlist1 [name=bvar1 value=102 index=16#1200
subindex=16#1 ] };
END_VAR
```

(b) via a declaration in the VAR_CONFIG Interface:

There is the possibility to create an entry for a variable in a parameter list of type 'Variables' by placing a pragma in a VAR_CONFIG window (independent of variable configurations, which also are done in the VAR_CONFIG interface): (If the parameter list does not yet exist, it will be created automatically)

Syntax: {parameter list=<name> path=<path> [<key>=<value> <key>=<value> ...further keys] }
 <path> Path of the variable, for which the entry should be generated, e.g. "PLC_PRG.act1.var_x"

Example: For variable var_x an entry is created in parameter list "varlist1", symbolic name is xvar".

```
VAR_CONFIG
{parameter list=varlist1 path=PLC_PRG.act1.var_x [ name=xvar ] }
END_VAR
```

2. Entries in parameter lists of type 'Template' via function blocks and structures

Pragmas in variable declarations in function blocks and structures can be used to create entries in parameter lists of type 'Template'. (If the template does not yet exist, it will be created automatically.)

Syntax: {template list=<name> [<key>=<value> <key>=<value> ...further keys] }

Example: Variable strvar, which is an element of structure "stru1", should be entered in a parameter list "templ1" of type 'Template'; symbolic name (member) of the entry is "struvar1, the access level is "low" :

```
TYPE stru :
STRUCT
ivar:INT;
strvar:STRING{template list=vorl1 [member=struvar1 accesslevel=low] };
END_STRUCT
END_TYPE
```

3. Entries in parameter lists of type 'Instance' (for arrays, function block- or structure variables)

(a) via declarations in programs or global variable lists

At the declaration of arrays, function block- or structure variables in a program or in a global variables list directly an parameter list of type 'Instance' can be created:

Syntax: {instance list=<name> template=<template> baseindex=<index> basesubindex=<subindex> [<key>=<value for first element > <key>=<value for first element > ...further keys for first element] | [<key>=<value for second element > <key>=<value for second element > ..further keys for second element] | [keys for further elements]}<key>=<value> <key>=<value> ...further keys] }

For arrays the key "template" will automatically be defined with the implicitly available template "ARRAY", for structures and function blocks an appropriate template must be available in the Parameter Manager and must be part of the here described declaration.

For each particular array- resp. structure or function block element an individual entry can be pre-defined in the parameter list: For example per each element an own definition [name=<elementname>] can be specified.

The key definitions of each particular element (all keys of a particular element are enclosed by the same square brackets!) re to be arranged in a row, separated by empty spaces. The definition

packages for the elements will automatically refer to the elements according to the ascending order of the index (Member). If there are not as many key-definitions available as there are elements in the array resp. the structure or the function block, then the remaining elements will get assigned the same values as the last individually defined element (Exception for key "name" for arrays, see below)! (see example 1b).

Automatisms concerning key "name" when entering arrays in parameter lists:

- If you do not specify a name for an array element in the pragma, that element and all following elements automatically will get the names <POU name>_<Array variable name>_<corresponding array index number>. Example: Array variable ARRVAR [1..8] of INT in POU PLC_PRG is to be entered in a variable list via a pragma; If key "name" is not defined in the pragma, the particular array elements in the parameter list automatically will be named "PLC_PRG_arrvar_1" to "PLC_PRG_arrvar_8".
- If you define any name "<name>_<first index number of array range>" for the first array element via the pragma, the following array elements in the parameter list automatically will be named "<name>_<corresponding index number>". Example: Array variable ARRVAR [1..8] of INT in is to be entered in a variable list via a pragma; The pragma just specifies [name=xyz_1]" for the first array element and thus in the parameter list the following elements automatically will get the names "xyz_2" to "xyz_8".

Attention: Do not define a value for the key "Member"; this column will be filled automatically by using the array index values.

Examples:

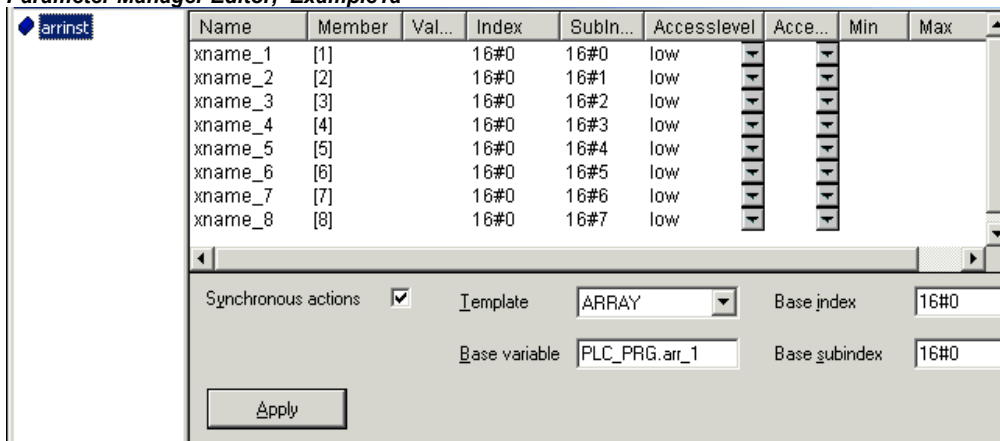
Example1a: An array variable arr_1 is declared as described in the following in order to create a parameter list arrinst of type 'Instance' (if not yet available), in which the array components get entered, whereby each element first will get a symbolic name xname_<index number> (could be edited afterwards in the parameter manager) and the subindex will be counted up by 1 for each entry, starting with 0 (basesubindex). Accesslevel=low will be applied to all elements.

```
arr_1: ARRAY [1..8] OF INT{instance list=arrinst template=ARRAY baseindex=16#0
baseSubindex=16#0 [name=xname_1 accesslevel=low] }
```

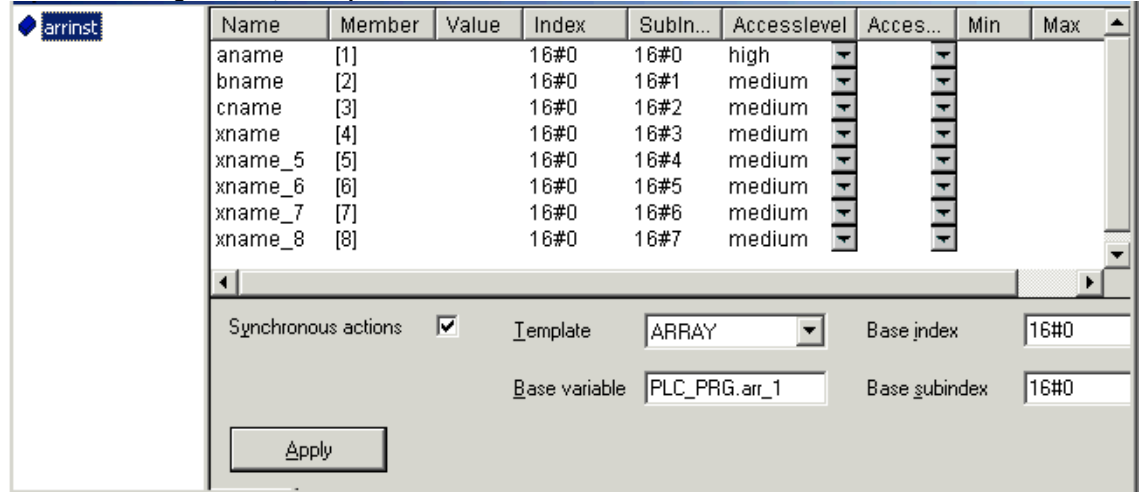
Example1b: For an array variable arr_1 just for the elements 1 to 4 already names are defined via the pragma, the elements 5 to 8 therefore automatically will get the name of element 4 extended by a Underline and the appropriate index number, i.e. "xname_5" to " xname_8". Regard that further key-definitions for a particular element must be entered within the same square brackets as shown here for the first and for the fourth element regarding the accesslevel:

```
arr_1: ARRAY [1..8] OF INT{instance list=arrinst template=ARRAY baseindex=16#0
baseSubindex=16#0[name=aname accesslevel=high] [name=bname] [name=cname]
[name=xname accesslevel=medium]};
```

Parameter Manager Editor, Example1a



Parameter Manager Editor, Example1b

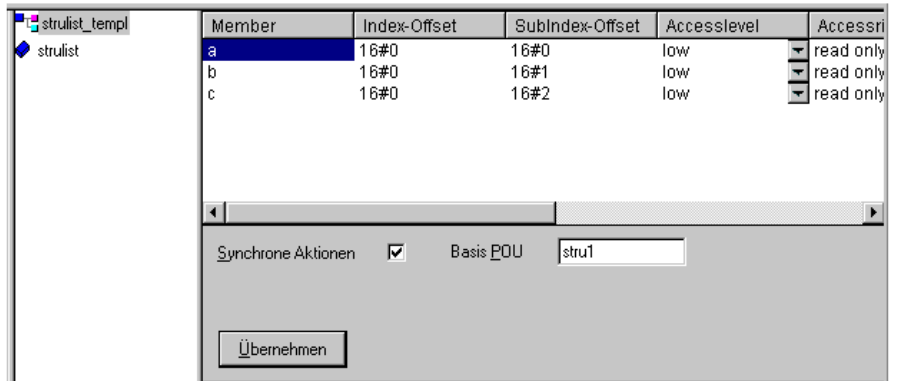
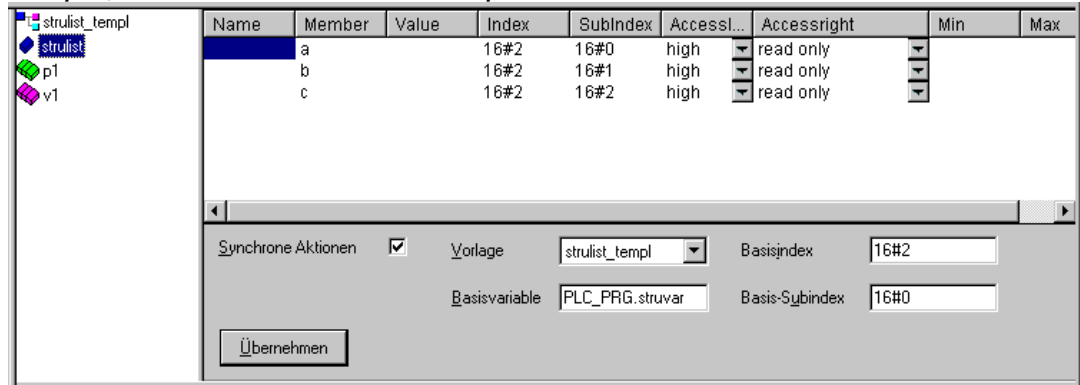


Example2:

A structure variable of type "stru1" (contains variables a,b,c) is declared as described in the following in order to get entries in a parameter list of type 'Instance' which is basing on the template "strulist_temp"; the list will get entries for the components a,b,c, no symbolic names are assigned, the access level is set to "high" and each index value defined by the template will be increased by 2. Make sure that the template defined in the pragma is available in the Parameter Manager:

```
struvar:stru1instance list=strulist template=strulist_temp baseindex=16#2
basesubindex=16#0 [accesslevel=high] ;
```

Example2, Entries for structure variable in Template



(b) via declarations in the VAR_CONFIG interface

For variables which can be instantiated, you can directly create entries in an parameter list of type 'Instance' by a pragma in a VAR_CONFIG window (independent of variable configuration definitions which are also done in this interface). If the template does not yet exist, it will be created automatically.

Make sure that the Template defined in the pragma is already available in the Parameter Manager.

Syntax: {instance list=<name> path=<path> template=<template> baseindex=<index> basesubindex=<subindex>[<key>=<value> <key>=<value> ...further keys] }

<path>: The instance path of the variable;e.g. "PLC_PRG.fb1inst", whereby fb1inst is an instance of function block fb1.

Example: The following entry in a VAR_CONFIG window will create entries for all variables of function block "fb1" in an instance list "varinst1" basing on the template "fb1_templ" (which must be already available). For each entry the index offset, which is predefined by the template, will be increased by 2 (baseindex), the subindex offset will not be modified (basesubindex). Each entry gets a symbolic name "fb1var", which you can edit afterwards in the Parameter Manager.

```
VAR_CONFIG
{instance list=varinst1 path=PLC_PRG.fb1 template=fb1_templ baseindex=16#2
basesubindex=16#0 [ name=fb1var ]}
END_VAR
```

Pragma for nonpersistent data types

Normally the following is valid: Even if only one local variable in a function block or a structure is declared persistent, at usage of an instance automatically all components will be stored in the persistent information (persist.dat) on the runtime system. In order to save space you can use the pragma

```
{nonpersistent}
```

in the declaration of the function block resp. the structure. It effects that only those components of the function block resp. structure, which are declared as "persistent", will be entered to the persistent info.

Example:

If an instance of the following function block is declared as persistent, only the variables local and fblevel3 will be written to the persistent info. Without pragma {nonpersistent} all function block variables would be stored there.

```
FUNCTION_BLOCK FB_Level_2
{nonpersistent}
VAR_INPUT
  bvar_in : BOOL;
END_VAR
VAR_OUTPUT
  bvar_out : BOOL;
END_VAR
VAR
  ivar2 : INT;
END_VAR
VAR PERSISTENT
  local : INT := 33;
  fblevel3 : FB_Level_3;
END_VAR
```

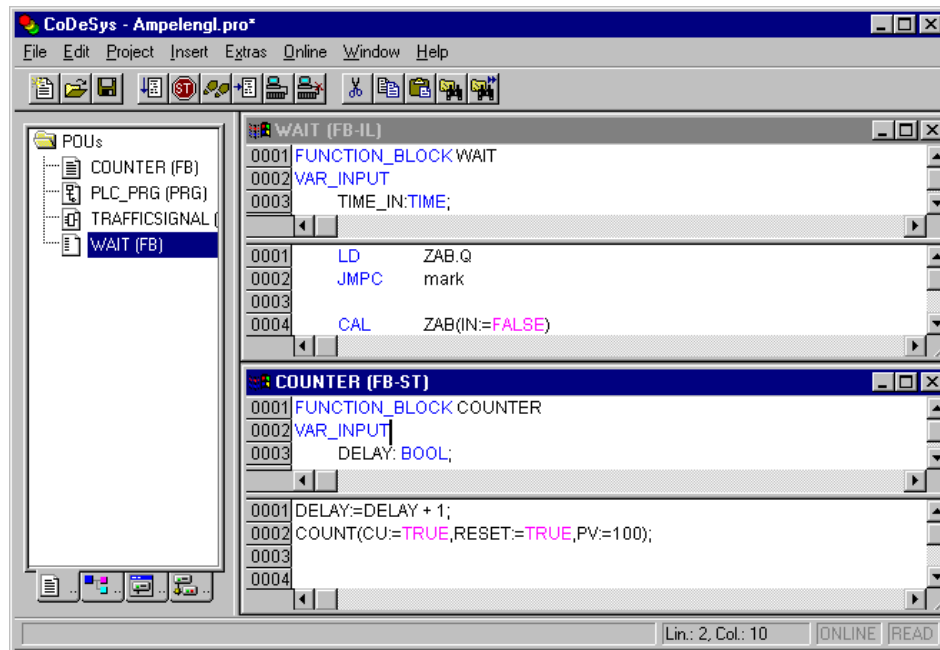
5.3 The Text Editors

5.3.1 Working in text editors

Text editors used for the implementation portion (the Instruction List editor and the Structured Text editor) of **CoDeSys** provide the usual Windows text editor functions.

The implementation in the text editors is supported by syntax coloring.

In Overwrite mode the status bar shows a black **OV**. You can switch between Overwrite mode and Insert mode by key <Ins>



The most important commands are found in the context menu (right mouse button or <Ctrl>+<F10>).

The text editors use the following menu commands in special ways:

'Insert' 'Operators' in text editors

With this command all of the operators available in the current language are displayed in a dialog box.

If one of the operators is selected and the list is closed with **OK**, then the highlighted operator will be inserted at the present cursor position. (This is managed here just as it is in the Input Assistant).

'Insert' 'Operand' in text editors

With this command all variables in a dialog box are displayed. You can select whether you would like to display a list of the global, the local, or the system variables.

If one of the operands is chosen, and the dialog box is closed with **OK**, then the highlighted operand will be inserted at the present cursor position. (This is managed here just as it is in the Input Assistant).

'Insert' 'Function' in text editors

With this command all functions will be displayed in a dialog box. You can choose whether to have a list displaying user-defined or standard functions.

If one of the functions is selected and the dialog box is closed with **OK**, then the highlighted function will be inserted at the current cursor position. (The management will proceed, as in the input selection.)

If the **With arguments** option was selected in the dialog box, then the necessary input and output variables will also be inserted.

'Insert' 'Function Block' in text editors

With this command all function blocks are displayed in a dialog box. You can choose whether to have a list displaying user-defined or standard function blocks.

If one of the function blocks is selected and the dialog box is closed with **OK**, then the highlighted function block will be inserted at the current cursor position. (This is managed here just as it is in the Input Assistant).

If the **With arguments** option was selected in the dialog box, then the necessary input variables of the function block will also be inserted. However you are not forced to assign these parameters.

Calling POU's with output parameters in text editors

The output parameters of a called POU can be directly assigned upon being called in the text languages IL and ST.

Example:

Output parameter out1 of afbinst is assigned variable a.

```
IL: CAL afbinst(in1:=1, out1=>a)
```

```
ST: afbinst(in1:=1, out1=>a);
```

If the POU is inserted via input assistant (<F2>) with option 'With arguments' in the implementation window of a ST or IL POU, it will automatically be displayed with all parameters in this syntax. However you are not forced to assign these parameters.

The text editors in Online mode

The online functions in the editors are set breakpoint and single step processing (steps). Together with the monitoring, the user thus has the debugging capability of a modern Windows standard language debugger.

In Online mode, the text editor window is vertically divided in halves. On the left side of the window you will then find the normal program text; on the right side you will see a display of the variables whose values were changed in the respective lines. The width of the window halves can be modified by drawing the divider with the mouse.

The display is the same as in the declaration part. That means that when the PLC is running, the present values of the respective variables will be displayed.

The following should be noted when monitoring expressions or Bit-addressed variables: in the case of expressions, the value of the entire expression is always displayed. Example: a AND b is displayed in blue or with „:=TRUE" if both a and b are TRUE. For Bit-addressed variables, the bit value that is addressed is always monitored (e.g. a.3 is displayed in blue or with „:=TRUE, if a has the value 4).

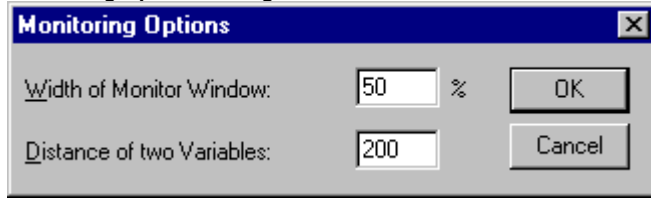
If you place the mouse pointer briefly above a variable, then the type, the address and the comment about the variable will be displayed in a Tooltip.

'Extras' 'Monitoring Options'

With this command you can configure your monitoring window. In the text editors, the window is divided into two halves during monitoring. The program is located in the left half. In the right half, all variables that are located in the corresponding program line are monitored.

You can specify the Monitor Window **Width** and which **Distance** two variables should have in a line. An distance declaration of 1 corresponds, in this case, to a line height in the selected font.

Regard that the width of the window halves also can be modified by drawing the divider with the mouse.

Monitoring Options Dialog Box

Breakpoint Positions in Text Editor

Breakpoint positions include all positions in a program at which values of variables can change or where the program flow branches off. (Exception: function calls. If necessary, a breakpoint in the function must be set here.) At the positions lying in between, a breakpoint would not even make sense, since nothing has been able to change in the data since the preceding breakpoint position.

This results in the following breakpoint positions in the IL:

At the start of the POU

At every LD, LDN (or, in case a LD is located at a label, then at the label)

At every JMP, JMPC, JMPCN

At every label

At every CAL, CALC, CALCN

At every RET, RETC, RETCN

At the end of the POU

Structured Text accommodates the following breakpoint positions:

At every assignment

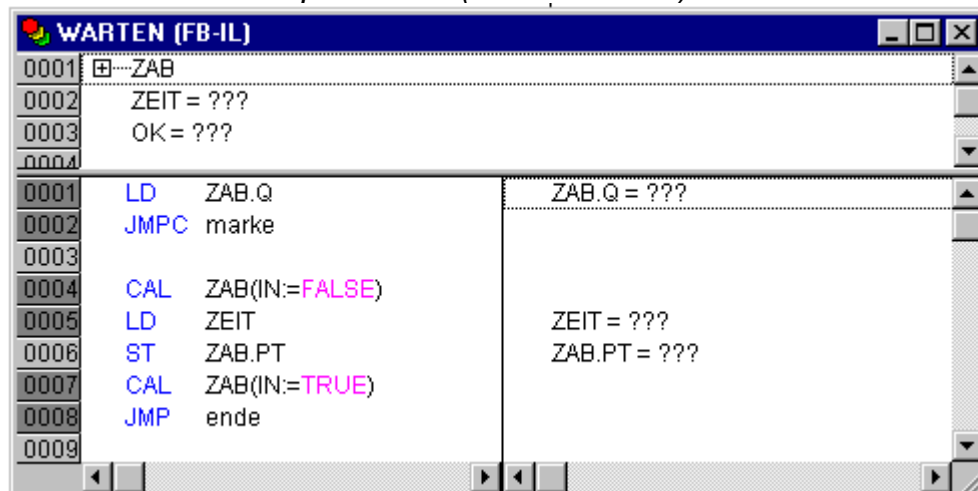
At every RETURN and EXIT instruction

in lines where conditions are being evaluated (WHILE, IF, REPEAT)

At the end of the POU

Breakpoint positions are marked by the display of the line number field in the color which is set in the project options.

IL Editor with Possible Breakpoint Positions (darker number fields)



How do you set a breakpoint?

In order to set a breakpoint, click the line number field of the line where you want to set a breakpoint. If the selected field is a breakpoint position, then the color of the line numbers field will change from dark gray to light blue, and the breakpoint will be activated in the PLC.

Deleting Breakpoints

Correspondingly, in order to delete a breakpoint, click on the line number field of the line with the breakpoint to be deleted.

Setting and deleting of breakpoints can also be selected via the menu ('Online' 'Toggle Breakpoint'), via the function key <F9>, or via the symbol in the tool bar.

What happens at a breakpoint?

If a breakpoint is reached in the PLC, then the screen will display the break with the corresponding line. The line number field of the line where the PLC is positioned will appear in red. The user program is stopped in the PLC.

If the program is at a breakpoint, then the processing can be resumed with 'Online' 'Run'.

In addition, with 'Online' 'Step over' or 'Step in' you can cause the program to run to the next breakpoint position. If the instruction where you are located is a CAL command, or, if there is a function call in the lines up to the next breakpoint position, then you can use '**Step over**' to bypass the function call. With '**Step in**', you will branch to the open POU

Line Number of the Text Editor

The line numbers of the text editor give the number of each text line of an implementation of a POU.

In Off-line mode, a simple click on a special line number will mark the entire text line.

In Online mode, the background color of the line number indicates the breakpoint status of every line. The standard settings for the colors are:

dark grey: This line is a possible position for a breakpoint.

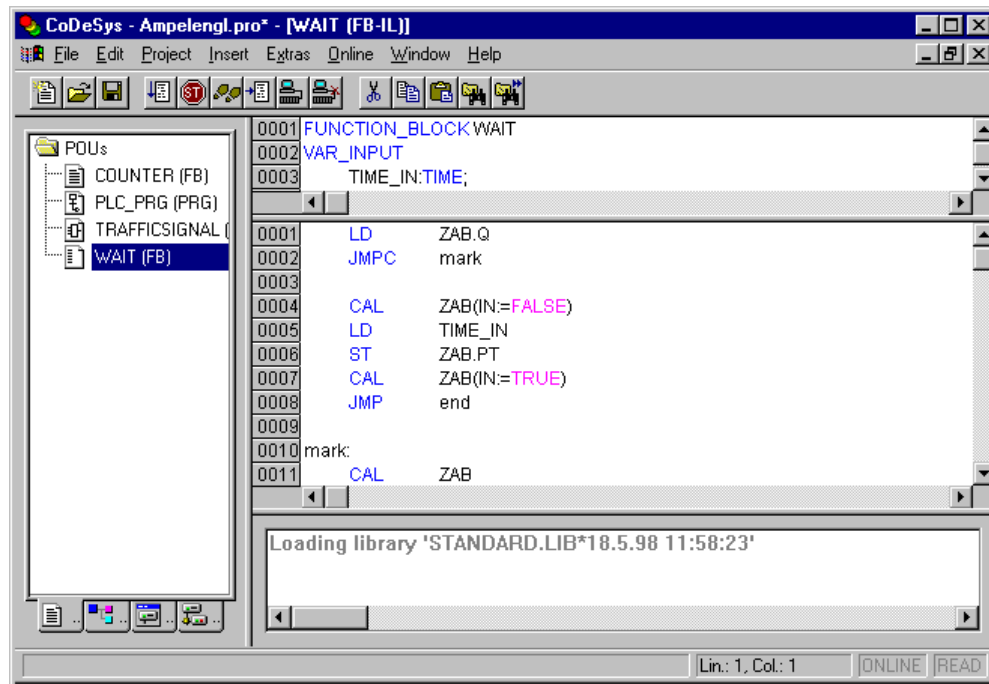
light blue: a breakpoint has been set in this line.

red: The program has reached this point.

In Online mode, simply clicking the mouse will change the breakpoint status of this line.

5.3.2 The Instruction List Editor

This is how a POU written in the IL looks under the corresponding **CoDeSys** editor:



All editors for POUs consist of a declaration part and a body. These are separated by a screen divider.

The Instruction List editor is a text editor with the usual capabilities of Windows text editors. The most important commands are found in the context menu (right mouse button or <Ctrl>+<F10>). Multiline POU calls are also possible:

Example:

```

CAL CTU inst(
CU:=%IXI0,
PV:=(
LD A
ADD 5
)
)

```

For information concerning the language, see Chapter 2.2.1, Instruction List (IL).

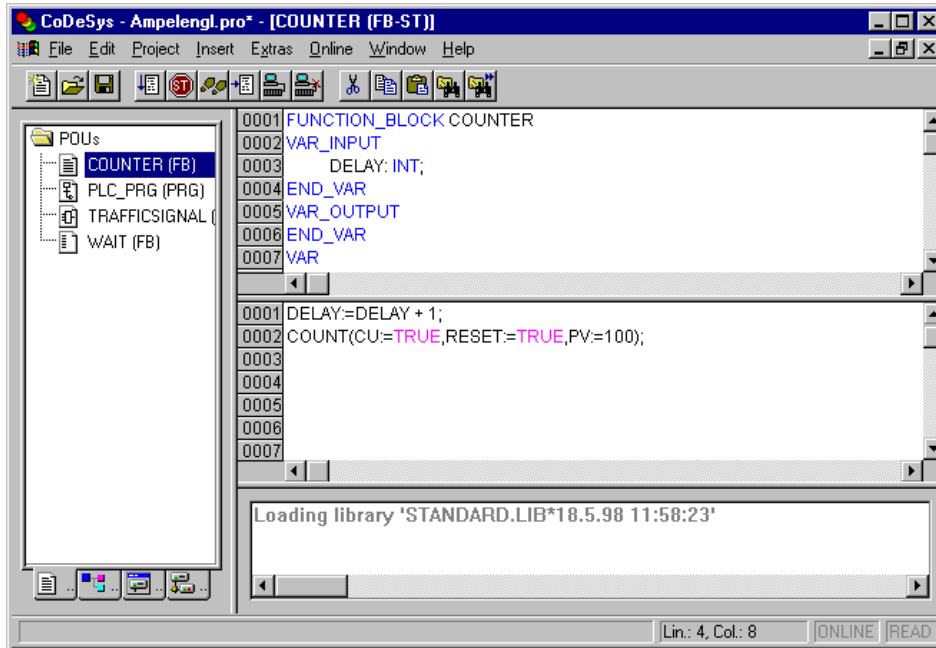
IL in Online mode

With the 'Online' 'Flow control' command, an additional field in which the accumulator contents is displayed is inserted in the IL editor on the left side of every line.

For further information concerning the IL editor in Online mode, see 'The Text Editors in Online Mode'.

5.3.3 The Editor for Structured Text

This is how a POU written in ST appears under the corresponding **CoDeSys** editor:



All editors for POUs consist of a declaration part and a body. These are separated by a screen divider.

The editor for Structured Text is a text editor with the usual capabilities of Windows text editors. The most important commands are found in the context menu (right mouse button or <Ctrl>+<F10>).

For information about the language, see Chapter 2.2.2, 'Structured Text (ST)'.

5.4 The Graphic Editors

5.4.1 Working in graphic editors

The editors of the graphically oriented languages, sequential function chart SFC, ladder diagram LD and function block diagram FBD and of free graphic function block diagrams have many points in common. In the following paragraphs these features will be summarized; the specific descriptions of LD, FBD and SFC, as well as the Sequential Function Chart language SFC follow in separate sections. The implementation in the graphics editors is supported by syntax coloring.

Zoom

Objects such as POUs, actions, transitions etc. in the languages SFC, LD, FBD, CFC and in visualizations can be enlarged or reduced in size with a zoom function. All elements of the window contents of the implementation part are affected; the declaration part remains unchanged.

In standard form, every object is displayed with the zoom level 100%. The zoom level that is set is saved as an object property in the project.

The printing of project documentation always occurs at the 100% display level!

The zoom level can be set through a selection list in the toolbar. Values between 25% and 400% can be selected; individual values between 10% and 500% can be entered manually.

The selection of a zoom level is only available if the cursor rests on an object created in a graphical language or a visualization object.

Even with the object zoomed, cursor positions in the editors can be further selected and even reached with the arrow keys. Text size is governed by the zoom factor and the font size that is set.

The execution of all editor menu features (e.g. inserting a box) as a function of cursor position is available at all zoom levels, taking the same into account.

In Online mode, each object is displayed according to the zoom level that has been set; Online functionality is available without restriction.

When the IntelliMouse is used, an object can be enlarged/reduced by pressing the <CTRL> key and at the same time turning the wheel forward or backwards.

Network

In the LD and FBD editors, the program is arranged in a list of networks. Each network is designated on the left side by a serial network number and has a structure consisting of either a logical or an arithmetic expression, a program, function or function block call, and a jump or a return instruction.

Label

Each network has a label that can optionally be left empty. This label is edited by clicking the first line of the network, directly next to the network number. Now you can enter a label, followed by a colon.

Comments, Networks with Linebreaks, 'Extras' 'Options'

Every network can be supplied with a multi-lined comment. In the dialog 'Function Block and Ladder Diagram Options', which can be opened by executing the command '**Extras**' '**Options**', you can do the settings concerning comments and linebreaks:

Dialog Function Block and Ladder Diagram Options

The screenshot shows the 'Function Block and Ladder Diagram Options' dialog box. It features a title bar with a close button (X). The main area contains several settings:

- Minimum Comment Size:** A text box containing '1' followed by 'Lines'.
- Maximum Comment Size:** A text box containing '5' followed by 'Lines'.
- Alternative Look & Feel for Ladder Diagram:** A group box containing:
 - Comments per Contact
 - Lines for Contact Comment:** A text box containing '2' followed by 'Lines'.
 - Lines for Variable Text:** A text box containing '1' followed by 'Lines'.
 - Networks with Linebreaks
 - Replace with Symbol after entering Address
 - Set Contact Comment to Symbol Comment
 - Show Address of Symbol
 - Show Variable Comments per Rung in Printout

Buttons for 'OK', 'Cancel', and 'Apply Options' are located on the right side of the dialog.

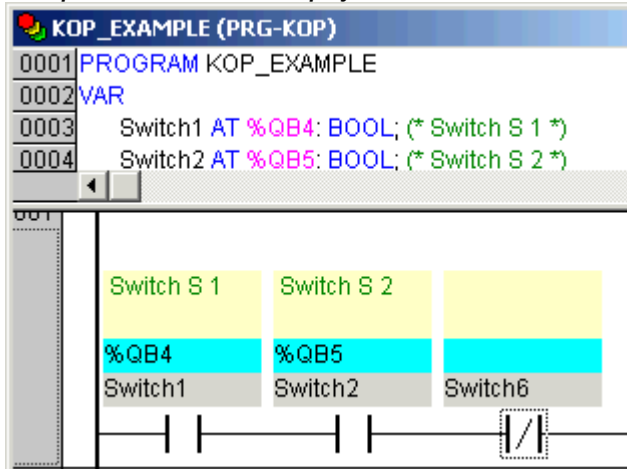
In the **Maximum Comment Size** field you can enter the maximum number of lines to be made available for a network comment (The default value here is 4.) In the field **Minimum Comment Size** you can enter the number of lines that generally should be reserved for . If, for example, the number 2 is entered, then, at the start of each network there will be two empty lines after the label line. The default value here is 0, which has the advantage of allowing more networks to fit in the screen area.

If the minimal comment size is greater than 0, then in order to enter a comment you simply click in the comment line and then enter the comment. Otherwise you first must select the network to which a comment is to be entered, and use **'Insert' 'Comment'** to insert a comment line. In contrast to the program text, comments are displayed in grey.

Alternative Look & Feel: The following options allow to define an alternative look of the networks.

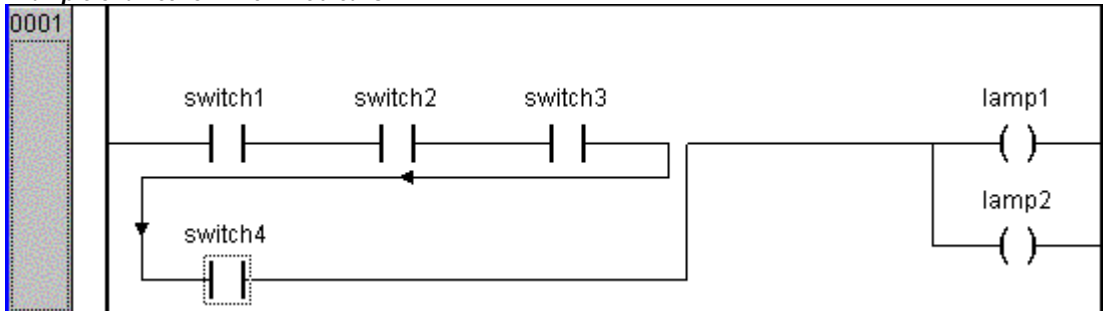
Comments per Contact (only for Ladder editor): If this option is activated, you can assign an individual comment to each contact or coil. In the edit field **Lines for Variable Comment** enter the number of lines which should be reserved and displayed for the comment. If this setting is done, a comment field will be displayed in the editor above each contact and coil where you can insert text. If 'Comments per Contact' is activated, then in the Ladder editor also the number of lines (**Lines for Variable Text:**) can be defined which should be used for the variable name of the contact resp. coil. This is used to display even long names completely by breaking them into several lines. In the following example 2 lines are defined for the variable comment and 1 line for the variable text:

Example of a network with display of variable comment and address:



Networks with Linebreaks (only for Ladder editor): If this option is activated, linebreaks will be forced in the networks as soon as the network length exceeds the given window size and some of the elements would not be visible.

Example of a network with linebreaks



Replace with Symbol after entering Address: (only for Ladder editor): If this option is activated, you can enter an address at a box resp. at a contact or coil (e.g. "%QB4") and this address will be replaced immediately by the name of the variable which is assigned to the address. If there is no variable assigned to the entered address, the address remains displayed unchangedly.

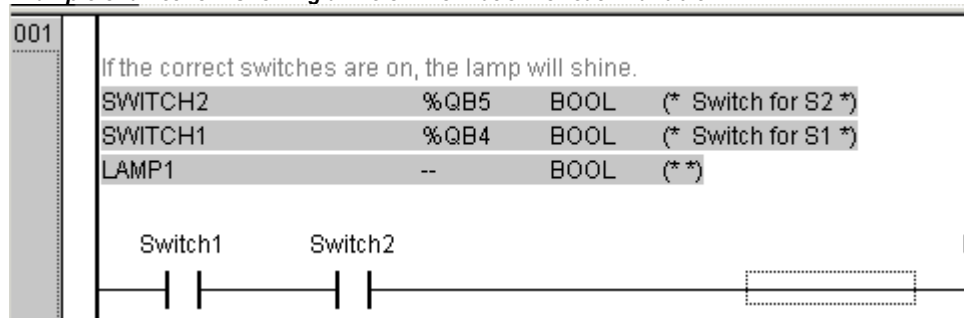
Set Contact Comment to Symbol Comment: If this option is activated, in the comment field of a contact resp. a coil that comment will be displayed which has been defined at the declaration of the variable used for the contact or coil (see above, picture at 'Comments per Contact'). The comment then can be edited.. For this purpose however the option 'Comments per Contact' also must be activated. Regard that a comment which has been entered already locally at a contact or coil will be

replaced automatically by the variable comment in any case, even if the variable has not got a comment in its declaration!

Show Address of Symbol: (only for Ladder editor): If this option is activated and a variable assigned to a contact or coil is assigned to an address, the address will be displayed above the variable name (see above, picture at 'Comments per Contact').

Show Variable Comments per Rung in Printout: If this option is activated, in each network for each variable used in that network there will be displayed a line showing the name, address, data type and comment for this variable, as defined in the variables declaration. This might be of useful for a documentation (printout) of the project.

Example of a network showing a line of information for each variable



Applying the options:

OK: Press this button to apply the settings on the actual POU and to close the options dialog.

Apply options: Press this button to apply the settings on the whole project. A dialog will open asking you where you have explicitly to confirm that you want to do that.

'Insert' 'Network (after)' or 'Insert' 'Network (before)'

Shortcut: <Shift>+<T> (Network after)

In order to insert a new network in the FBD or the LD editor, select the '**Insert' 'Network (after)'** or **the 'Insert' 'Network (before)'** command, depending on whether you want to insert the new network before or after the present network. The present network can be changed by clicking the network number. You will recognize it in the dotted rectangle under the number. With the <Shift key> and a mouse click you can select from the entire area of networks, from the present one to the one clicked.

The network editors in the online mode

In the FBD and the LD editors you can only set breakpoints for networks. The network number field of a network for which a breakpoint has been set, is displayed in blue. The processing then stops in front of the network, where the breakpoint is located. In this case, the network number field is displayed in red. With single step processing (steps), you can jump from network to network.

All values are monitored upon entering and exiting network POUs (Program Organization Units).

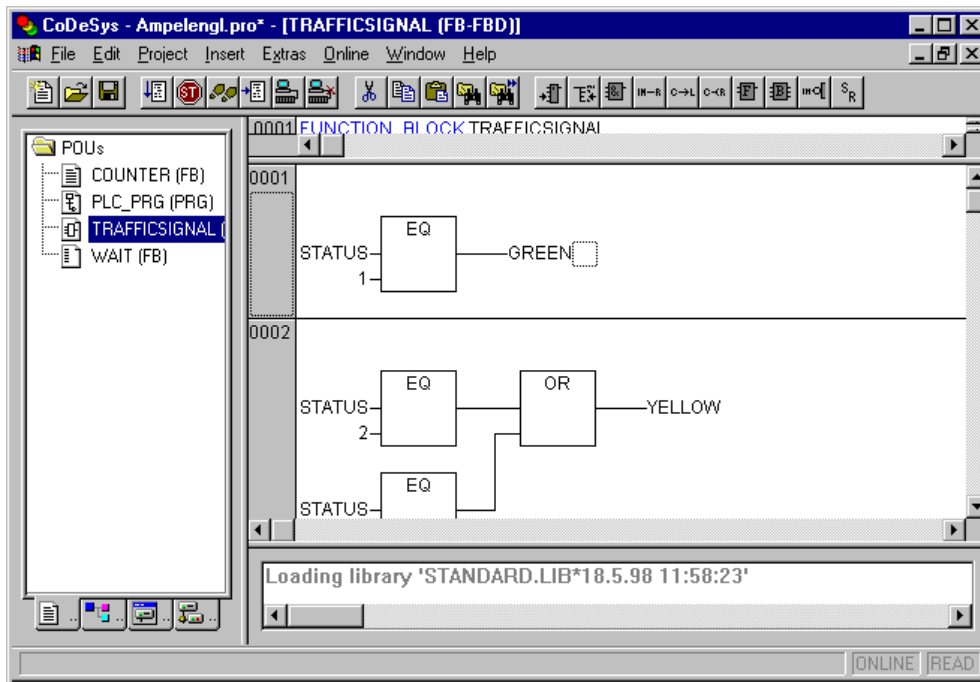
The following should be noted when monitoring expressions or Bit-addressed variables: In expressions, e.g. a AND b, used as transition condition or function block input, the value of the whole expression is always displayed (a AND b is shown in blue or as :=TRUE, if a and b are TRUE). For Bit-addressed variables, the bit value that is addressed is always monitored (e.g. a.3 is displayed in blue or with „:=TRUE, if a has the value 4)

The flow control is run with the 'Online' 'Flow control' command. Using the flow control, you can view the present values that are being carried in the networks over the connecting lines. If the connecting lines do not carry Boolean values, then the value will be displayed in a specially inserted field. The monitor fields for variables that are not used (e.g. in the function SEL) are displayed in a shade of grey. If the lines carry Boolean values, then they will be shaded blue, in the event that they carry TRUE. Therefore, you can accompany the flow of information while the PLC is running.

If you place the mouse pointer briefly above a variable, then the type, the address and the comment about the variable will be displayed in a Tooltip.

5.4.2 The Function Block Diagram Editor

This is how a POU written in the FBD under the corresponding **CoDeSys** editor looks:



The Function Block Diagram editor is a graphic editor. It works with a list of networks, in which every network contains a structure that displays, respectively, a logical or an arithmetical expression, the calling up of a function block, a function, a program, a jump, or a return instruction. The most important commands are found in the context menu (right mouse button or <Ctrl>+<F10>).

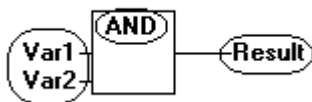
Regard the possibility to switch the display of a FBD-POU between FBD- and KOP editor, in offline mode as well as in online mode. Furtheron regard the possibilities offered by the 'Function Block and Ladder Diagram Options' (see Chapter 5.4.1: Comments, Linebreaks, 'Extras' 'Options').

Cursor positions in FBD

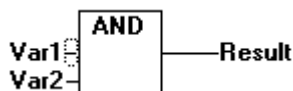
Every text is a possible cursor position. The selected text is on a blue background and can now be changed.

You can also recognize the present cursor position by a dotted rectangle. The following is a list of all possible cursor positions with an example:

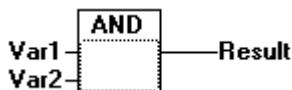
- 1) Every text field (possible cursor positions framed in black):



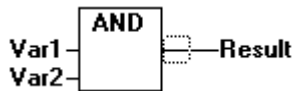
- 2) Every input:



- 3) Every operator, function, or function block:



4) Outputs, if an assignment or a jump comes afterward:



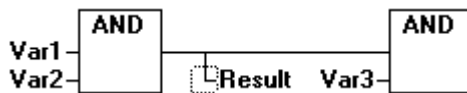
5) The lined cross above an assignment, a jump, or a return instruction:



6) Behind the outermost object on the right of every network ("last cursor position," the same cursor position that was used to select a network):



7) The lined cross directly in front of an assignment:



How to set the cursor in FBD

The cursor can be set at a certain position by clicking the mouse, or with the help of the keyboard.

Using the arrow keys, you can jump to the nearest cursor position in the selected direction at any time. All cursor positions, including the text fields, can be accessed this way. If the last cursor position is selected, then the <up> or <down> arrow keys can be used to select the last cursor position of the previous or subsequent network.

An empty network contains only three question marks "???". By clicking behind these, the last cursor position is selected.

'Insert' 'Assign' in FBD

Symbol:  **Shortcut: <Ctrl>+<A>**

This command inserts an assignment.

Depending on the selected position (see 'Cursor positions in FBD'), insertion takes place directly in front of the selected input (Cursor Position 2), directly after the selected output (Cursor Position 4) or at the end of the network (Cursor Position 6).

For an inserted assignment, a selection can be made accompanying the entered text "???", and the assignment can be replaced by the variable that is to be assigned. For this you can also use the Input Assistant. For the possibility to enter an address instead of the variable name please regard the description of the options dialog in Chapter 5.4.1: Comments, Linebreaks, 'Extras' 'Options'.

In order to insert an additional assignment to an existing assignment, use the **'Insert' 'Output'** command.

'Insert' 'Jump' in FBD

Symbol:  **Shortcut: <Ctrl>+<L>**

This command inserts a jump.

Depending on the selected position (see 'Cursor positions in FBD'), insertion takes place directly in front of the selected input (Cursor Position 2), directly after the selected output (Cursor Position 4) or at the end of the network (Cursor Position 6).

For an inserted jump, a selection can be made accompanying the entered text "???", and the jump can be replaced by the label to which it is to be assigned.

'Insert' 'Return' in FBD

Symbol:  **Shortcut:** <Ctrl>+<R>

This command inserts a RETURN instruction.

Depending on the selected position (see 'Cursor positions in FBD'), insertion takes place directly in front of the selected input (Cursor Position 2), directly after the selected output (Cursor Position 4), directly before the selected line cross (Cursor Position 5), or at the end of the network (Cursor Position 6)

'Insert' 'Box' in FBD

Symbol:  **Shortcut:** <Ctrl>+

With this command, operators, functions, function blocks and programs can be inserted. First of all, it is always inserted an "AND" operator. This can be converted by Selection and Overwrite of the type text („AND") into every other operator, into every function, into every function block and every program. You can select the desired POU by using Input Assistant (<F2>). If the new selected block has another minimum number of inputs, these will be attached. If the new block has a smaller highest number of inputs, the last inputs will be deleted.

In functions and function blocks, the formal names of the in- and outputs are displayed.

In function blocks there exists an editable instance field above the box. If another function block that is not known is called by changing the type text, an operator box with two inputs and the given type is displayed. If the instance field is selected, Input Assistant can be obtained via <F2> with the categories for variable selection.

The newest POU is inserted at the selected position:

If an input is selected (Cursor Position 2), then the POU is inserted in front of this input. The first input of this POU is linked to the branch on the left of the selected input. The output of the new POU is linked to the selected input.

If an output is selected (Cursor Position 4), then the POU is inserted after this output. The first input of the POU is connected with the selected output. The output of the new POU is linked to the branch with which the selected output was linked.

If a POU, a function, or a function block is selected (Cursor Position 3), then the old element will be replaced by the new POU.

As far as possible, the branches will be connected the same way as they were before the replacement. If the old element had more inputs than the new one, then the unattachable branches will be deleted. The same holds true for the outputs.

If a jump or a return is selected, then the POU will be inserted before this jump or return. The first input of the POU is connected with the branch to the left of the selected element. The output of the POU is linked to the branch to the right of the selected element.

If the last cursor position of a network is selected (Cursor Position 6), then the POU will be inserted following the last element. The first input of the POU is linked to the branch to the left of the selected position.

All POU inputs that could not be linked will receive the text "???". This text must be clicked and changed into the desired constant or variable.

If there is a branch to the right of an inserted POU, then the branch will be assigned to the first POU output. Otherwise the outputs remain unassigned.

'Insert' 'Input'


Symbol:  **Shortcut: <Ctrl>+<U>**

This command inserts an operator input. With many operators, the number of inputs may vary. (For example, ADD can have 2 or more inputs.)

In order to extend such an operator by an input, you need to select the input in front of which you wish to insert an additional input (Cursor Position 1); or you must select the operator itself (Cursor Position 3), if a lowest input is to be inserted (see 'Cursor positions in FBD').

The inserted input is allocated with the text "???". This text must be clicked and changed into the desired constant or variable. For this you can also use the Input Assistant. For the possibility to enter an address instead of the variable name please regard the description of the options dialog in Chapter 5.4.1: Comments, Linebreaks, 'Extras' 'Options'.

'Insert' 'Output'

Symbol: 

This command inserts an additional assignment into an existing assignment. This capability serves the placement of so-called assignment combs; i.e., the assignment of the value presently located at the line to several variables.

If you select the lined cross above an assignment (Cursor Position 5) (see 'Cursor positions in FBD') or the output directly in front of it (Cursor Position 4), then there will be another assignment inserted after the ones already there.

If the line cross directly in front of an assignment is selected (Cursor Position 4), then another assignment will be inserted in front of this one.

The inserted output is allocated with the text "???". This text must be clicked and changed into the desired variable. For this you can also use the Input Assistant. For the possibility to enter an address instead of the variable name please regard the description of the options dialog in Chapter 5.4.1: Comments, Linebreaks, 'Extras' 'Options'.

'Extras' 'Negate'

Symbol:  **Shortcut: <Ctrl>+<N>**

With this command you can negate the inputs, outputs, jumps, or RETURN instructions. The symbol for the negation is a small circle at a connection.

If an input is selected (Cursor Position 2) (see 'Cursor positions in FBD'), then this input will be negated.

If an output is selected (Cursor Position 4), then this output will be negated.

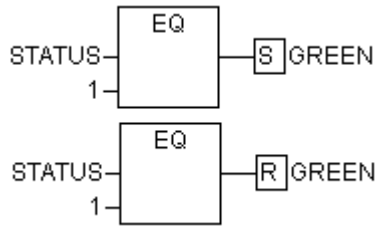
If a jump or a return is marked, then the input of this jump or return will be negated.

A negation can be canceled through renewed negation.

'Extras' 'Set/Reset'

Symbol: 

With this command you can define outputs as Set or Reset Outputs. A grid with Set Output is displayed with [S], and a grid with Reset Output is displayed with [R].

Set/Reset Outputs in FBD

An *Output Set* is set to TRUE, if the grid belonging to it returns TRUE. The output now maintains this value, even if the grid jumps back to FALSE.

An *Output Reset* is set to FALSE, if the grid belonging to it returns FALSE. The output maintains its value, even if the grid jumps back to FALSE.

With multiple executions of the command, the output will alternate between set, reset, and normal output.

'Extras' 'View'

Using this command for a POU created in the FBD-Editor you can choose, whether it should be displayed in the LD- (ladder logic) or in the FBD-Editor (Function block diagram). This is possible in offline as well as in online mode.

Open instance

This command corresponds to the 'Project' 'Open instance' command.

It is available in the context menu (<F2>) or in the 'Extras' menu, if the cursor is positioned on the name of a function block in a text editor or if the function block box is selected in a graphic editor.

Cutting, Copying, Pasting, and Deleting in FBD

The commands used to **'Cut'**, **'Copy'**, **'Paste'**, and **'Delete'** are found under the **'Edit'** menu item.

If a line cross is selected (Cursor Position 5) (see 'Cursor positions in FBD'), then the assignments, jumps, or RETURNS located below the crossed line will be cut, deleted, or copied.

If a POU is selected (Cursor Position 3), then the selected object itself, will be cut, deleted, or copied, along with all of the branches dependent on the inputs, with the exception of the first (highest position) branch.

Otherwise, the entire branch located in front of the cursor position will be cut, deleted, or copied.

After copying or cutting, the deleted or copied part is located on the clipboard and can now be pasted, as desired.

In order to do so, you must first select the pasting point. Valid pasting points include inputs and outputs.

If a POU has been loaded onto the clipboard (As a reminder: in this case all connected branches except the first are located together on the clipboard), the first input is connected with the branch before the pasting point.

Otherwise, the entire branch located in front of the pasting point will be replaced by the contents of the clipboard.

In each case, the last element pasted is connected to the branch located in front of the pasting point.

Note: The following problem is solved by cutting and pasting: A new operator is inserted in the middle of a network. The branch located on the right of the operator is now connected with the first input, but should be connected with the second input. You can now select the first input and perform the command **'Edit' 'Cut'**. Following this, you can select the second input and perform the command **'Edit' 'Paste'**. This way, the branch is dependent on the second input.

The Function Block Diagram in the Online Mode

In the Function Block Diagram, breakpoints can only be set to networks. If a breakpoint has been set to a network, then the network numbers field will be displayed in blue. The processing then stops in front of the network where the breakpoint is located. In this case, the network numbers field will become red. Using stepping (single step), you can jump from network to network.

The current value is displayed for each variable. Exception: If the input to a function block is an expression, only the first variable in the expression is monitored.

Doubleclicking on a variable opens the dialog box for writing a variable. Here it is possible to change the present value of the variable. In the case of Boolean variables, no dialog box appears; these variables are toggled.

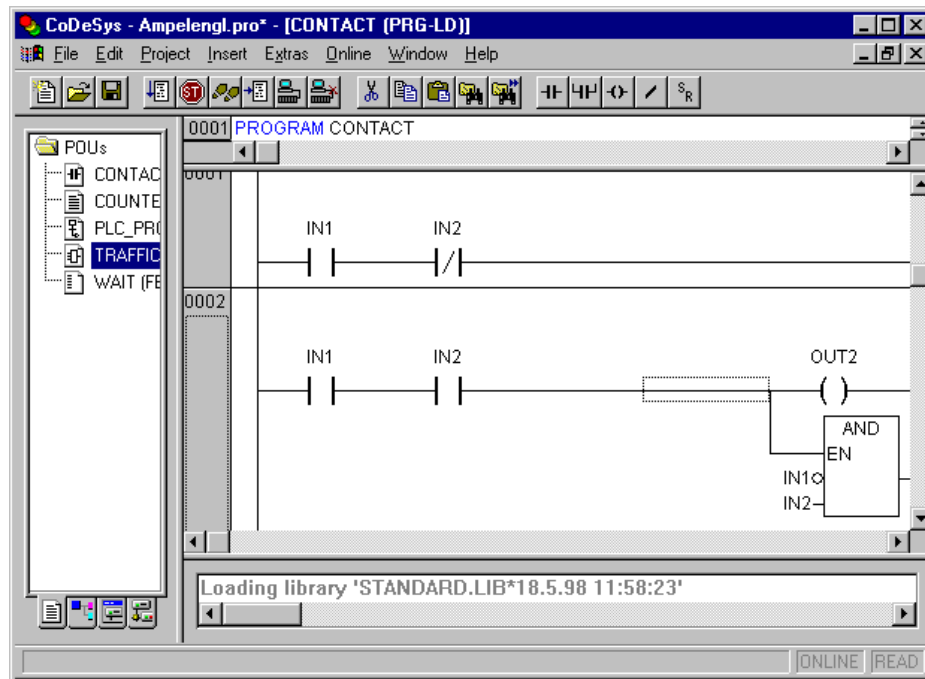
The new value will turn red and will remain unchanged. If the 'Online' 'Write values' command is given, then all variables are placed in the selected list and are once again displayed in black.

The flow control is started with the 'Online' 'Flow control' command. Using the flow control, you can view the present values that are being carried in the networks over the connecting lines. If the connecting lines do not carry Boolean values, then the value will be displayed in a specially inserted field. If the lines carry Boolean values, then they will be shaded blue in the event that they carry TRUE. By this means, you can accompany the flow of information while the PLC is running.

If you place the mouse pointer briefly above a variable, then the type, the address and the comment about the variable will be displayed in a Tooltip.

5.4.3 The Ladder Editor

This is how a POU written in the LD appears in the **CoDeSys** editor:



All editors for POU's consist of a declaration part and a body. These are separated by a screen divider.

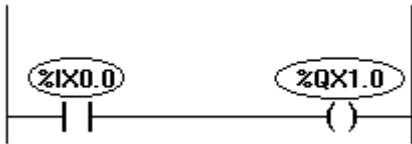
The LD editor is a graphic editor. The most important commands are found in the context menu (right mouse button or <Ctrl>+<F10>).

For information about the elements, see Chapter 2.2.6, Ladder Diagram (LD). Regard the possibilities concerning comments, address input and alternative look&feel given by the options dialog (see Chapter 5.4.1).

Cursor Positions in the LD Editors

The following locations can be cursor positions, in which the function block and program accessing can be handled as contacts. POU with EN inputs and other POU connected to them are treated the same way as in the Function Block Diagram. Information about editing this network part can be found in chapter 5.4.2, FBD Editor.

1. Every text field (possible cursor positions framed in black)



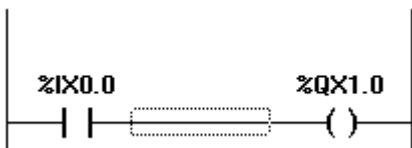
2. Every Contact or Function Block



3. Every Coil



4. The Connecting Line between the Contacts and the Coils.



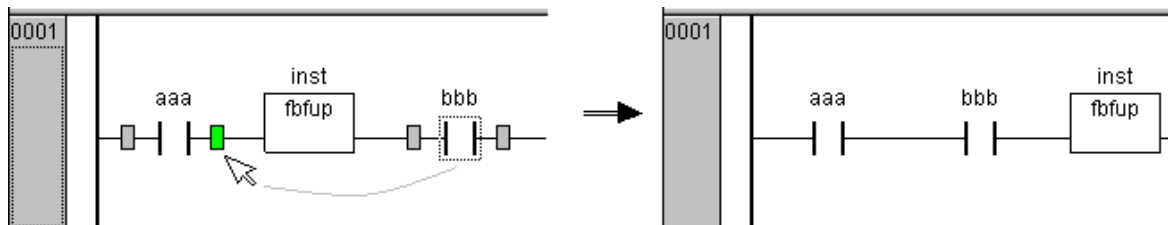
The Ladder Diagram uses the following menu commands in a special way:

Move elements or names in the LD-Editor

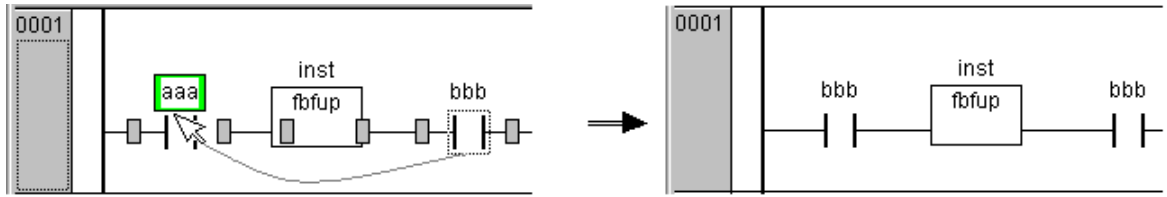
An element (contact, coil, function block) or just the name of an element (variable name, address, comment) can be moved to a different position within a LD POU by "drag&drop".

In order to do this select the desired element (contact, coil, function block) and drag it - keeping the mouse key pressed - away from the current position. Thereupon all possible positions within all networks of the POU, to which the element might be moved, will be indicated by grey-filled rectangles.

Move the element to one of these positions and let off the mouse key: the element will be inserted at the new position.



If you however move the element to the name (variable name) of another element, the name field will be shaded green. If you then let off the mouse key, the previous name will be replaced by the "dragged" one. If additionally address and comment are displayed (options), the copying also will apply to those.



'Insert' 'Network (before)' in LD

Symbol:

This command inserts a network in the Ladder editor. If there are already networks, the new one will be inserted before the currently focused.

'Insert' 'Network (after)' in LD

Symbol:

This command inserts a network in the Ladder editor. If there are already networks, the new one will be inserted after the currently focused.

'Insert' 'Contact' in LD

Symbol: Shortcut: <Ctrl>+<K>

Use this command in the LD editor in order to insert a contact in front of the marked location in the network.

If the marked position is a coil or the connecting line between the contacts and the coils, then the new contact will be connected serially to the previous contact connection.

The contact is preset with the text "???". You can click on this text and change it to the desired variable or the desired constant. For this you can also use the Input Assistant.

Regard the possibility of entering an address instead of the variable name, if this is configured appropriately in the 'Function Block and Ladder Diagram Options' (see Chapter 5.4.1: Comments, Linebreaks, 'Extras' 'Options').

Also regard the option Networks with linebreaks, which you also can activate in the Ladder Diagram Options.

'Insert' 'Contact (negated)' in LD

Symbol: Shortcut: <Ctrl> + <G>

This command inserts a negated contact. The same is true as for the commands 'Insert' 'Contact' and 'Extras' 'Negate', which in combination also could be used to insert a negated contact.

'Insert' 'Parallel Contact' in LD

Symbol: Shortcut: <Ctrl>+<R>

Use this command in the LD editor to insert a contact parallel to the marked position in the network.

If the marked position is a coil or the connection between the contacts and the coils, then the new contact will be connected in parallel to the entire previous contact connection.

The contact is preset with the text "???". You can click on this text and change it to the desired variable or the desired constant. For this you can also use the Input Assistant.

For the possibility of entering addresses, using linebreaks in variable names and using comments per contact please see the description of the 'Function Block and Ladder Diagram Options' (see Chapter 5.4.1: Comments, Linebreaks, 'Extras' 'Options').

'Insert' 'Parallel Contact (negated)' in LD

Symbol:  Shortcut: <Ctrl> + <D>

This command inserts a negated parallel contact. The same is true as for the commands 'Insert' 'Parallel Contact' and 'Extras' 'Negate', which in combination also could be used to insert a negated parallel contact.

'Insert' 'Coil' in LD

Symbol:  Shortcut: <Ctrl>+<L>

You can use this command in the LD editor to insert a coil in parallel to the previous coils.

If the marked position is a connection between the contacts and the coils, then the new coil will be inserted as the last. If the marked position is a coil, then the new coil will be inserted directly above it.

The coil is given the text "???" as a default setting. You can click on this text and change it to the desired variable. For this you can also use the Input Assistant.

For the possibility of entering addresses, using linebreaks in variable names and using comments per contact (coil, function block) please see the description of the 'Function Block and Ladder Diagram Options' (see Chapter 5.4.1: Comments, Linebreaks, 'Extras' 'Options').

'Insert' "Set' coil' in LD

Symbol:  Shortcut: <Ctrl> + <I>

This command inserts a set coil. The same is true as for the commands 'Insert' 'Coil' and 'Extras' 'Set/Reset', which in combination also could be used to get a set coil.

'Insert' 'Reset' coil' in LD

Symbol: 

This command inserts a reset coil. The same is true as for the commands 'Insert' 'Coil' and 'Extras' 'Set/Reset', which in combination also could be used to get a reset coil.

'Insert' 'Function Block' in LD

Symbol:  Shortcut: <Ctrl>+

Use this command in order to insert an operator, a function block, a function or a program as a POU. For this, the connection between the contacts and the coils, or a coil, must be marked. The new POU at first has the designation AND. If you wish, you can change this designation to another one. For this you can also use the Input Assistant. Both standard and self-defined POUs are available.

The first input to the POU is placed on the input connection, the first output on the output connection; thus these variables must definitely be of type BOOL. All other in- and outputs of the POU are filled with the text „???". These prior entries can be changed into other constants, variables or addresses. For this you can also use the Input Assistant.

For the possibility of entering addresses, using linebreaks in variable names and using comments per contact (coil, function block) please see the description of the 'Function Block and Ladder Diagram Options' (see Chapter 5.4.1: Comments, Linebreaks, 'Extras' 'Options').

POUs with EN Inputs in LD

If you want to use your LD network as a PLC for calling up other POUs, then you must merge a POU with an EN input. Such a POU is connected in parallel to the coils. Beyond such a POU you can develop the network further, as in the Function Block Diagram. You can find the commands for insertion at an EN POU under the menu item **'Insert' 'Insert at Blocks'**.

An operator, a function block, a program or a function with EN input performs the same way as the corresponding POU in the Function Block Diagram, except that its execution is controlled on the EN

input. This input is annexed at the connecting line between coils and contacts. If this connection carries the information "On", then the POU will be evaluated.

If a POU has been created once already with EN input, then this POU can be used to create a network. This means that data from usual operators, functions, and function blocks can flow in an EN POU and an EN POU can carry data to such usual POUs.

If, therefore, you want to program a network in the LD editor, as in FBD, you only need first to insert an EN operator in a new network. Subsequently, from this POU, you can continue to construct from your network, as in the FBD editor. A network thus formed will perform like the corresponding network in FBD.

'Insert' 'Box with EN' in LD

Symbol: 

Use this command to insert a function block, an operator, a function or a program with EN input into a LD network.

The marked position must be the connection between the contacts and the coils (Cursor Position 4) or a coil (Cursor Position 3). The new POU is inserted in parallel to the coils and underneath them; it contains initially the designation "AND". If you wish, you can change this designation to another one. For this you can also use the Input Assistant.

'Insert' 'Insert at Blocks' in LD

With this command you can insert additional elements into a POU that has already been inserted (also a POU with EN input). The commands below this menu item can be executed at the same cursor positions as the corresponding commands in the Function Block Diagram.

With **Input** you can add a new input to the POU.

With **Output** you can add a new output to the POU.

With **POU**, you insert a new POU. The procedure is similar to that described under 'Insert' 'POU'.

With **Assign** you can insert an assignment to a variable. At first, this is shown by three question marks „???", which you edit and replace with the desired variable. Input assistance is available for this purpose.

'Insert' 'Rising edge detection' in LD

Symbol: 

This command inserts a R_TRIG function block, which serves to detect a rising edge (FALSE -> TRUE) at the incoming signal. The same is true as for command 'Insert' 'Function Block' which can be used to insert any available function block.

'Insert' 'Falling edge detection' in LD

Symbol: 

This command inserts a F_TRIG function block, which serves to detect a falling edge (TRUE -> FALSE) at the incoming signal. The same is true as for command 'Insert' 'Function Block' which can be used to insert any available function block.

'Insert' 'Timer (TON)' in LD

Symbol: 

This command inserts a timer function block of type TON. This serves to get a turn-on delay (delayed passing on of the incoming signal). For the inserting the same is true as for command 'Insert' 'Function Block', which also could be used to insert a TON module.

'Insert' 'Jump' in LD

With this command you can insert a parallel jump in the selected LD editor, in parallel, at the end of the previous coils. If the incoming line delivers the value "On", then the jump will be executed to the indicated label.

The marked position must be the connection between the contacts and the coils or a coil.

The jump is present with the text "???". You can click on this text and make a change in the desired label.

'Insert' 'Return' in LD

In the LD editor, you can use this command to insert a Return instruction in parallel at the end of the previous coils. If the incoming line delivers the value "On," then the processing of the POU in this network is broken off.

The marked position must be the connection between the contacts and the coils or a coil.

'Extras' 'Paste after' in LD

Use this command in the LD editor to paste the contents of the clipboard as serial contact below the marked position. This command is only possible if the contents of the clipboard and the marked position are networks comprised of contacts.

'Extras' 'Paste below' in LD

Use this command in the LD editor to insert the contents of the clipboard as parallel contact below the marked position. This command is only possible if the contents of the clipboard and the marked position are networks comprised of contacts.

'Extras' 'Paste above' in LD

Use this command in the LD editor to insert the contents of the clipboard as parallel contact above the marked position. This command is only possible if the contents of the clipboard and the marked position are networks comprised of contacts.

'Extras' 'Negate' in LD

Symbol:  **Shortcut:** <Ctrl>+<N>

Use this command to negate a contact, a coil, a jump or return instruction, or an input or output of EN POUs at the present cursor position.

Between the parentheses of the coil or between the straight lines of the contact, a slash will appear ((/) or (/)). If there are jumps, returns, or inputs or outputs of EN POUs, a small circle will appear at the connection, just as in the FBD editor.

The coil now writes the negated value of the input connection in the respective Boolean variable. Right at this moment, a negated contact switches the status of the input to the output, if the respective Boolean variable carries the value FALSE.

If a jump or a return is marked, then the input of this jump or return will be negated.

A negation can be cancelled through renewed negation.

'Extras' 'Set/Reset' in LD

Symbol: 

If you execute this command on a coil, then you will receive a Set Coil. Such a coil never overwrites the value TRUE in the respective Boolean variable. This means that once you have set the value of this variable to TRUE, it will always remain at TRUE. A Set Coil is designated with an "S" in the coil symbol.

If you execute this command once again, then you will be given a Reset coil. Such a coil never overwrites the value FALSE in the respective Boolean variable. This means that once you have set

the value of this variable to FALSE, it will always remain at FALSE. A Reset Coil is designated with an "R" in the coil symbol.

If you execute this command repeatedly, the coil will alternate between set, reset and normal coil.

The Ladder Diagram in the Online Mode

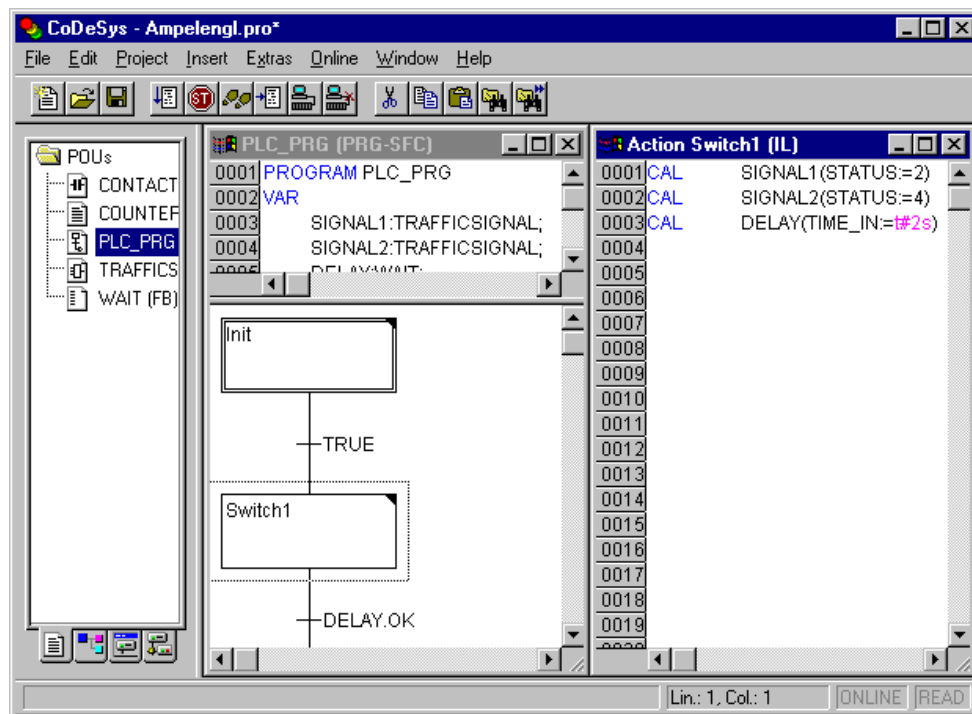
In Online mode, the contacts and coils in the Ladder Diagram that are in the "On" state are colored blue. Likewise, all lines over which the "On" is carried are also colored blue. At the inputs and outputs of function blocks, the values of the corresponding variables are indicated.

Breakpoints can only be set on networks; by using stepping, you can jump from network to network.

If you place the mouse pointer briefly above a variable, then the type, the address and the comment about the variable will be displayed in a Tooltip.

5.4.4 The Sequential Function Chart Editor

This is how a POU written in the SFC appears in the **CoDeSys** editor:



All editors for POU's consist of a declaration part and a body. These are separated by a screen divider.

The Sequential Function Chart editor is a graphic editor. The most important commands are found in the context menu (right mouse button or <Ctrl><F10>). Tooltips show in Offline as well as in Online mode and in the zoomed state the full names or expressions of steps, transitions, jumps, jump labels, qualifiers or associated actions.

For information about the Sequential Function Chart see Chapter 2.2.3, 'Sequential Function Chart'.

The editor for the Sequential Function Chart must agree with the particulars of the SFC. In reference to these, the following menu items will be of service.

Marking Blocks in the SFC

A marked block is a bunch of SFC elements that are enclosed in a dotted rectangle.

You can select an element (a step, a transition, or a jump) by pointing the mouse on this element and pressing the left mouse button, or you can use the arrow keys. In order to mark a group of several elements, press <Shift> for a block already marked, and select the element in the lower left or right

corner of the group. The resulting selection is the smallest cohesive group of elements that includes both of these elements.

Please regard, that a step can only be deleted together with the preceding or the succeeding transition!

'Insert' 'Step Transition (before)'

Symbol:  **Shortcut:** <Ctrl>+<T>

This command inserts a step in the SFC editor followed by a transition in front of the marked block.

You can select and replace the automatically specified step name „Step_<x>“ by another string, also the transition name.

Note: When renaming a step regard that no comment may be added. Exampe: „Step_xy (* counter *)“ is not allowed!

'Insert' 'Step Transition (after)'

Symbol:  **Shortcut:** <Ctrl>+<E>

This command inserts a step in the SFC editor followed by a transition after the first transition in the marked block. Concerning step name and transition name please see above 'Insert' 'Step-Transition (before)'.

Delete Step and Transition

A step can only be deleted together with the preceding or the succeeding transition. For this purpose put a selection frame around step and transition and choose command 'Edit' 'Delete' or press the key.

'Insert' 'Alternative Branch (right)'

Symbol:  **Shortcut:** <Ctrl>+<A>

This command inserts an alternative branch in the SFC editor as a right branch of the marked block. For this the marked block must both begin and end with a transition. The new branch is then made up of one transition.

'Insert' 'Alternative Branch (left)'

Symbol: 

This command inserts an alternative branch in the SFC editor as the left branch of the marked block. For this the marked block must both begin and end with a transition. The new branch is then made up of one transition.

'Insert' 'Parallel Branch (right)'

Symbol:  **Shortcut:** <Ctrl>+<L>

This command inserts a parallel branch in the SFC editor as the right branch of the marked block. For this the marked block must both begin and end with a step. The new branch is then made up of one step. To allow jumps to the parallel branches that have been created, these must be provided with a jump label.

'Insert' 'Parallel Branch (left)'

Symbol: 

This command inserts a parallel branch in the SFC editor as the left branch of the marked block. For this the marked block must both begin and end with a step. The new branch is then made up of one step. To allow jumps to the parallel branches that have been created, these must be provided with a jump label (see 'Extras' 'Add label to parallel branch').

'Insert' 'Jump'

This command inserts a jump in the SFC editor at the end of the branch, to which the marked block belongs. For this the branch must be an alternative branch.

The inserted text string 'Step' in the inserted jump can then be selected and replaced by the step name or the jump label of a parallel branch to be jumped to.

Concerning the step name please see above 'Insert' 'Step-Transition (before)'.

'Insert' 'Transition-Jump'

Symbol: 

This command inserts a transition in the SFC editor, followed by a jump at the end of the selected branch. For this the branch must be a parallel branch.

The inserted text string 'Step' in the inserted jump can then be selected and replaced by the step name or the jump label of a parallel branch to be jumped to.

Concerning step name and transition name please see above 'Insert' 'Step-Transition (before)'.

'Insert' 'Add Entry-Action'

With this command you can add an entry action to a step. An entry-action is only executed once, right after the step has become active. The entry-action can be implemented in a language of your choice.

A step with an entry-action is designated by an "E" in the bottom left corner.

'Insert' 'Add Exit-Action'

With this command you can add an exit-action to a step. An exit-action is only executed once, before the step is deactivated. The exit-action can be implemented in a language of your choice.

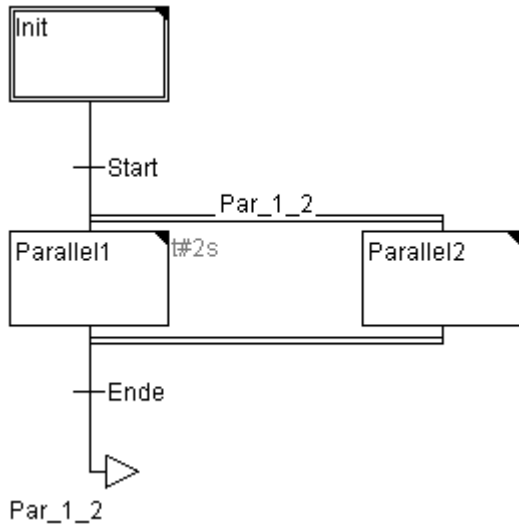
A step with an exit-action is designated by an "X" in the lower right corner.

'Extras' 'Paste Parallel Branch (right)'

This command pastes the contents of the clipboard as a right parallel branch of the marked block. For this the marked block must both begin and end with a step. The contents of the clipboard must, likewise, be an SFC block that both begins and ends with a step.

'Extras' 'Add label to parallel branch'

In order to provide a newly inserted parallel branch with a jump label, the transition occurring before the parallel branching must be marked and the command 'Add label to parallel branch' must be executed. At that point, the parallel branch will be given a standard name consisting of „Parallel" and an appended serial number, which can be edited according to the rules for identifier names. In the following example, "Parallel" was replaced by "Par_1_2" and the jump to the transition "End" was steered to this jump label.



Delete a label

A jump label can be deleted by deleting the label name.

'Extras' 'Paste after'

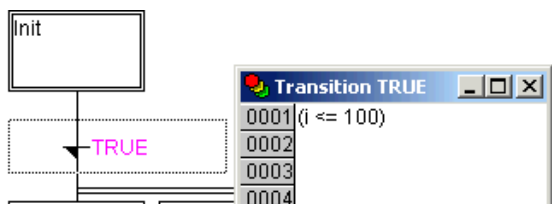
This command pastes the SFC block on the clipboard after the first step or the first transition of the marked block. (Normal copying pastes it in front of the marked block.) This will now be executed, if the resulting SFC structure is correct, according to the language norms.

'Extras' 'Zoom Action/Transition'

Shortcut: <Alt>+<Enter>

The action of the first step of the marked block or the transition body of the first transition of the marked block is loaded into the editor in the respective language, in which it has been written. If the action or the transition body is empty, then the language must be selected, in which it has been written.

Regard that the transition condition which is written within the editor window will take precedence over a condition which might be written directly at the transition mark. Example: If here $i > 100$, then the transition condition will be FALSE, although TRUE has been entered at the mark!



'Extras' 'Clear Action/Transition'

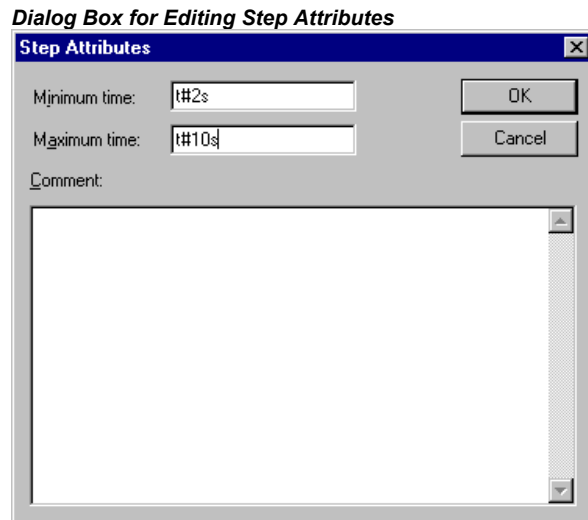
With this command you can delete the actions of the first step of the marked block or of the transitions body of the first transition.

If, during a step, you implement either only the action, the entry-action, or the exit-action, then the same will be deleted by the command. Otherwise a dialog box appears, and you can select which action or actions are to be deleted.

If the cursor is located in the action of an IEC step, then only this association will be deleted. If an IEC step with an associated action is selected, then this association will be deleted. During an IEC step with several actions, a selection dialog box will appear.

'Extras' 'Step Attributes'

With this command you can open a dialog box in which you can edit the attributes for the marked step.

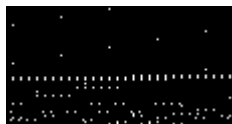


You can take advantage of three different entries in the step attribute dialog box. Under **Minimum Time**, you enter the minimum length of time that the processing of this step should take. Under the **Maximum Time**, you enter the maximum length of time that the processing of this step should take. Note that the entries are of the **TIME** type, so you use a TIME constant (i.e. T#3s) or a variable of the TIME type.

Under **Comment** you can insert a comment to the step. In the 'Sequential function chart options' dialog which you open under 'Extras' 'Options', you can then define whether comments, the time setting or nothing is displayed for the steps in the SFC editor. On the right, next to the step, either the comment or the time setting or none of both will appear.

Those attributes which are not displayed because of the options settings, can – additionally to the step name - be displayed in a tooltip, which appears when the cursor is placed on the upper resp. lower right corner of the step box.

If the maximum time is exceeded, SFC flags are set which the user can query.



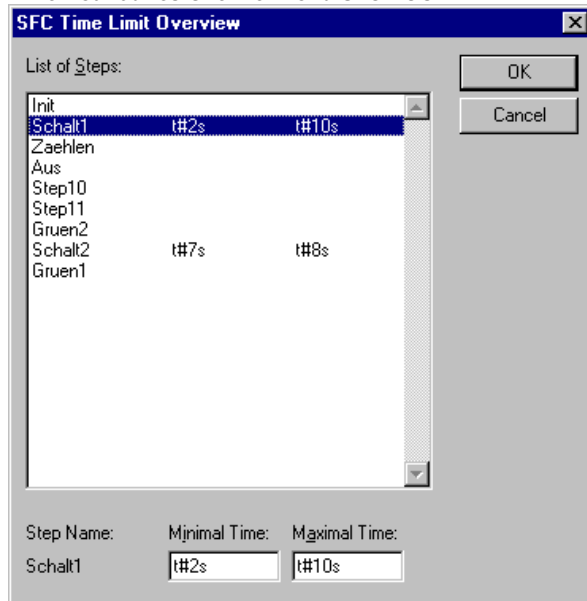
The example shows a step whose execution should last at least two, and at the most, ten seconds. In Online mode, there is, in addition to these two times, a display of how long the step has already been active.

'Extras' 'Time Overview'

With this command you can open a window in which you can edit the time settings of your SFC steps.

In the time boundaries overview, all steps of your SFC POU are displayed. If you have entered a time boundary for a step, then the time boundary is displayed to the right of the step (first, the lower limit, then the upper limit). You can also edit the time boundaries. To do so, click on the desired step in the overview. The name **of the step** is then shown below in the window. Go to the **Minimum Time** or **Maximum Time** field, and enter the desired time boundary there. If you close the window with **OK**, then all of the changes will be stored.

Time Boundaries Overview for a SFC POU



In the example, steps 2 and 6 have a time boundary. Shift1 lasts at least two, and at most, ten seconds. Shift2 lasts at least seven, and at most, eight seconds.

'Extras' 'Options'

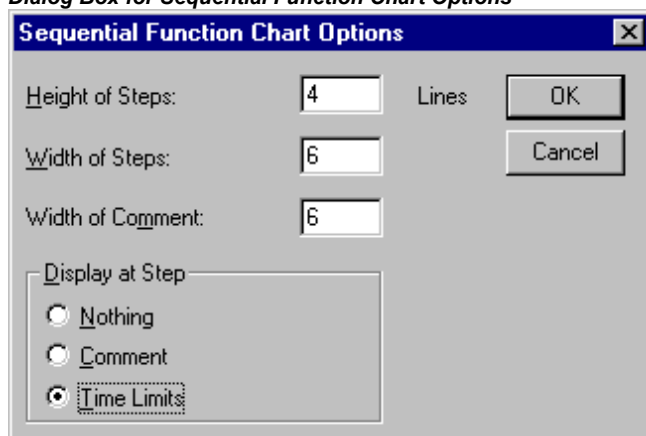
With this command you open a dialog box in which you can set different options for your SFC POU.

Under **Step Height**, you can enter how many lines high an SFC step can be in your SFC editor. 4 is the standard setting here. Under **Step Width**, you can enter how many columns wide a step should be. 6 is the standard setting here.

You can also preset the Display at Step, i.e. which of the attributes, defined 'Extras' 'Step Attributes' should be displayed next to the step. Either choose Comment, Time Limits.or Nothing:

- If "Nothing" is set, the defined comment and the time limits nevertheless can be shown in a tooltip, which appears when the cursor is placed on the lower right corner of the step box.
- If "Comment" is set, the defined comment and the time limits can be shown in a tooltip, which appears when the cursor is placed on the upper right corner of the step box.
- If "Time Limits" is set, the defined comment and the time limits nevertheless can be shown in a tooltip, which appears when the cursor is placed on the lower right corner of the step box.

Dialog Box for Sequential Function Chart Options



'Extras' 'Associate Action'

With this command actions and Boolean variables can be associated with IEC steps.

To the right of, and next to the IEC step, an additional divided box is attached, for the association of an action. It is preset in the left field with the qualifier "N" and the name "Action." Both presets can be changed. For this you can use the Input Assistant.

Maximum nine actions can be assigned to an IEC step.

New actions for IEC steps are created in the Object Organizer for an SFC POU with the **'Project' 'Add Action'** command.

'Extras' 'Use IEC-Steps'

Symbol: 

If this command is activated (denoted by a check in front of the menu item and a printed symbol in the Tool bar), then IEC steps will be inserted instead of the simplified steps upon insertion of step transitions and parallel branches.

If this option is switched on, the Init step is set as an IEC step when you create a new SFC POU.

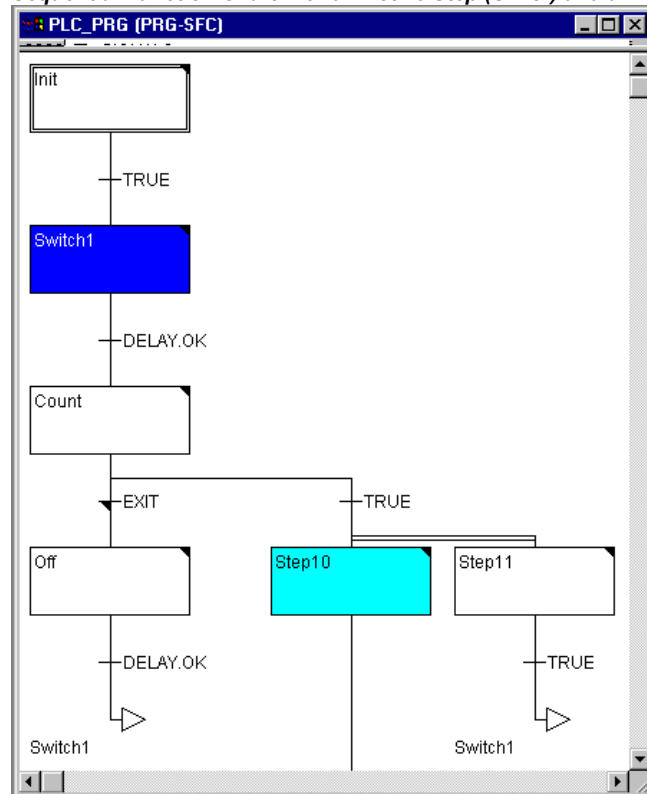
This settings are saved in the file "**CoDeSys.ini**" and are restored when **CoDeSys** gets started again.

Sequential Function Chart in Online Mode

With the Sequential Function Chart editor in Online mode, the currently active steps will be displayed in blue. If you have set it under 'Extras' 'Options', then the time management is depicted next to the steps. Under the lower and upper bounds that you have set, a third time indicator will appear from which you can read how long the step has already been active.



Sequential Function Chart with an Active Step (Shift1) and a Breakpoint (Step10)



In the picture above the step depicted has already been active 8 seconds and 410 milliseconds. The step must, however, be active for at least 7 minutes before the step will be left.

With **'Online' 'Toggle Breakpoint'** a breakpoint can be set on a step, or in an action at the locations allowed by the language in use. Processing then stops prior to execution of this step or before the location of the action in the program. Steps or program locations where a breakpoint is set are marked in light blue.

If several steps are active in a parallel branch, then the active step whose action will be processed next is displayed in red.

If IEC steps have been used, then all active actions in Online mode will be displayed in blue.

With the command **'Online' 'Step over'** it is stepped always to the next step which action is executed. If the current location is:

a step in the linear processing of a POU or a step in the rightmost parallel branch of a POU, execution returns from the SFC POU to the caller. If the POU is the main program, the next cycle begins.

a step in a parallel branch other than the rightmost, execution jumps to the active step in the next parallel branch.

the last breakpoint location within a 3S action, execution jumps to the caller of the SFC.

the last breakpoint location within an IEC action, execution jumps to the caller of the SFC.

the last breakpoint position within an input action or output action, execution jumps to the next active step.

With **'Online' 'Step in'** even actions can be stepped into. If an input, output or IEC action is to be jumped into, a breakpoint must be set there. Within the actions, all the debugging functionality of the corresponding editor is available to the user.

If you rest the mouse cursor for a short time on a variable in the declaration editor, the type, the address and the comment of the variable will be displayed in a **tooltip**

Please regard: If you rename a step and perform an Online Change while this step is active, the program will be stopped in undefined status !

Processing order of elements in a sequence:

1. First, all Action Control Block flags in the IEC actions that are used in this sequence are reset (not, however, the flags of IEC actions that are called within actions).
2. All steps are tested in the order which they assume in the sequence (top to bottom and left to right) to determine whether the requirement for execution of the output action is provided, and this is executed if that is the case.
3. All steps are tested in the order which they assume in the sequence to determine whether the requirement for the input action is provided, and this is executed if that is the case.
4. For all steps, the following is done in the order which they assume in the sequence:
 - If applicable, the elapsed time is copied into the corresponding step variable.
 - If applicable, any timeout is tested and the SFC error flags are serviced as required.
 - For non-IEC steps, the corresponding action is now executed.
5. IEC actions that are used in the sequence are executed in alphabetical order. This is done in two passes through the list of actions. In the first pass, all the IEC actions that are deactivated in the current cycle are executed. In the second pass, all the IEC actions that are active in the current cycle are executed.
6. Transitions are evaluated: If the step in the current cycle was active and the following transition returns TRUE (and if applicable the minimum active time has already elapsed), then the following step is activated.

The following must be noted concerning implementation of actions:

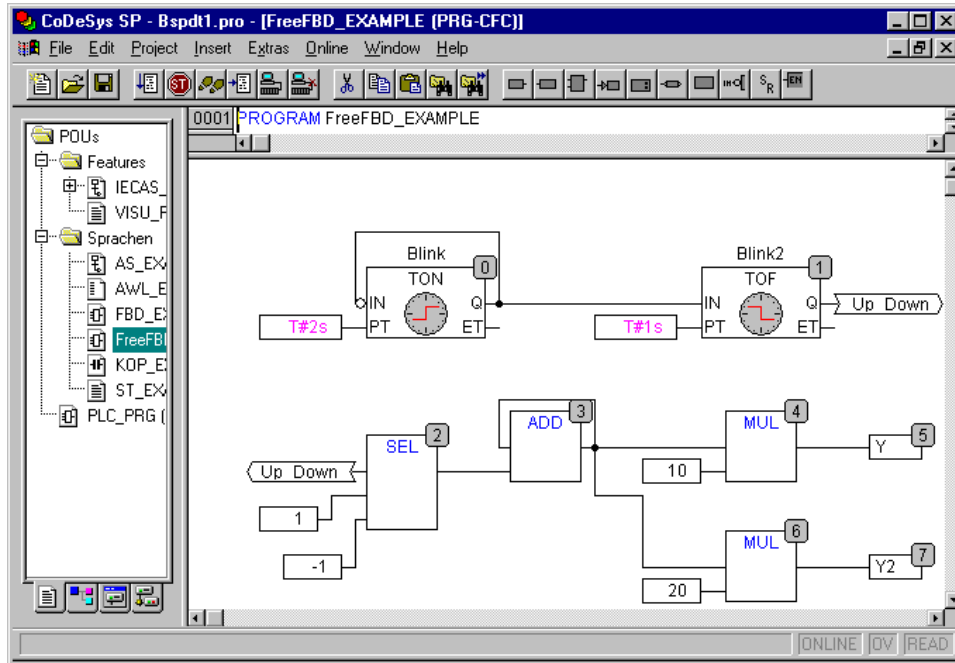
It can come about that an action is carried out several times in one cycle because it is associated with multiple sequences. (For example, an SFC could have two IEC actions A and B, which are both implemented in SFC, and which both call IEC action C; then in IEC actions A and B can both be active in the same cycle and furthermore in both actions IEC action C can be active; then C would be called twice).

If the same IEC action is used simultaneously in different levels of an SFC, this could lead to undesired effects due to the processing sequence described above. For this reason, an error message is issued in this case. It can possibly arise during processing of projects created with older versions of **CoDeSys**.

Note: In monitoring expressions (e.g. A AND B) in transitions, only the „Total value“ of the transition is displayed.

5.4.5 The Continuous Function Chart Editor (CFC)

It looks like a block which has been produced using the continuous function chart editor (CFC):



No snap grid is used for the continuous function chart editor so the elements can be placed anywhere. Elements of the sequential processing list include boxes, input, output, jump, label, return and comments. The inputs and outputs of these elements can be connected by dragging a connection with the mouse. The connecting line will be drawn automatically. The shortest possible connection line is drawn taking into account existing connections. The connecting lines are automatically adjusted when the elements are moved. If the case arises where a connecting line cannot be drawn simply because of lack of space, a red line will be shown between the input and the associated output instead. This line will be converted into a connecting line just as soon as space is available.

One advantage of the continuous function chart as opposed to the usual function block diagram editor FBD is the fact that feedback paths can be inserted directly.

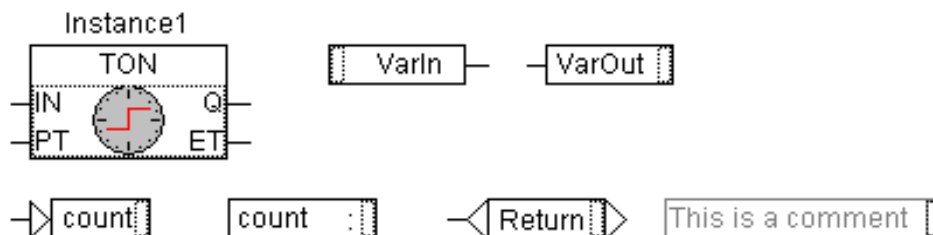
The most important commands can be found in the context menu

Cursor positions in the CFC

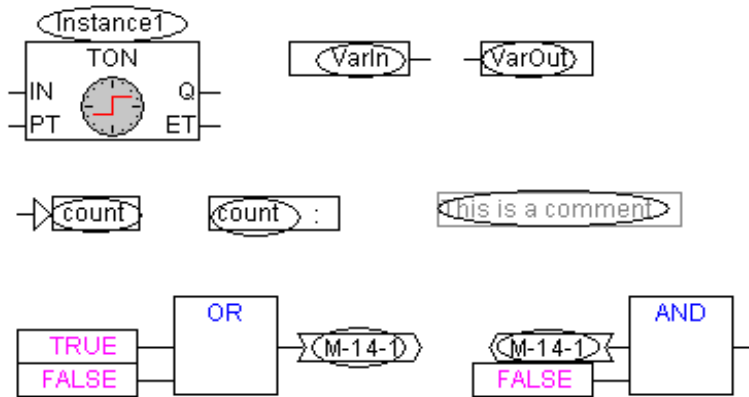
Each text is a possible cursor position. The selected text is shaded in blue and can be modified.

In all other cases the current cursor position is shown by a rectangle made up of points. The following is a list of all possible cursor positions with examples:

1. Trunks of the elements box, input, output, jump, label, return and comments.



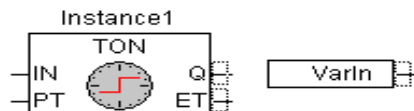
2. Text fields for the elements box, input, output, jump, label, return and comments as well as text fields for connection marker



3. Inputs for the elements box, input, output, jump and return



4. Outputs for the elements box and input:



'Insert' 'Box' in the CFC

Symbol: Shortcut: <Ctrl>+

This command can be used to paste in operators, functions, function blocks and programs First of all, it is always inserted an "AND" operator. This can be converted by Selection and Overwrite of the text into every other operator, into every function, into every function block and every program. The input assistance serves to select the desired block from the list of supported blocks. If the new block has another minimum number of inputs, these will be attached. If the new block has a smaller highest number of inputs, the last inputs will be deleted.

'Insert' 'Input' in CFC

Symbol: Shortcut: <Ctrl> + <E>

This command is used to insert an input. The text offered "???" can be selected and replaced by a variable or constant. The input assistance can also be used here.

'Insert' 'Output' in CFC

Symbol: Shortcut: <Ctrl>+<A>

This command is used to insert an output. The text offered "???" can be selected and replaced by a variable. The input assistance can also be used here. The value which is associated with the input of the output is allocated to this variable.

'Insert' 'Jump' in CFC

Symbol: Shortcut: <Ctrl>+<J>

This command is used to insert a jump. The text offered "???" can be selected and replaced by the jump label to which the program should jump.

The jump label is inserted using the command **'Insert 'Label'**.

'Insert' 'Label' in CFC

Symbol:  **Shortcut:** <Ctrl>+<L>

This command is used to insert a label. The text offered "???" can be selected and replaced by the jump label. In Online mode a RETURN label for marking the end of POU is automatically inserted.

The jump is inserted using the command **'Insert 'Jump'**.

'Insert' 'Return' in CFC

Symbol:  **Shortcut:** <Ctrl> + <R>

This command is used to insert a RETURN command. Note that in Online mode a jump label with the name RETURN is automatically inserted in the first column and after the last element in the editor; in stepping, it is automatically jumped to before execution leaves the POU.

'Insert' 'Comment' in CFC

Symbol:  **Shortcut:** <Ctrl> + <K>

This command is used to insert a comment.

You obtain a new line within the comment with <Ctrl> + <Enter>.

'Insert' 'Input of box' in CFC

Shortcut: <Ctrl> + <U>

This command is used to insert an input at a box. The number of inputs is variable for many operators (e.g. ADD can have two or more inputs).

To increase the number of inputs for such an operator by one, the box itself must be selected

Insert' 'In-Pin' in CFC, 'Insert' 'Out-Pin'

Symbol:  

These commands are available as soon as a macro is opened for editing. They are used for inserting in- or out-pins as in- and outputs of the macro. They differ from the normal in- and outputs of POUs by the way they are displayed and in that they have no position index.

'Extras' 'Negate' in CFC

Symbol:  **Shortcut:** <Ctrl> + <N>

This command is used to negate inputs, outputs, jumps or RETURN commands. The symbol for the negation is a small cross on the connection.

The input of the element block, output, jump or return is negated when it is selected.

The output of the element block or input is negated when it is selected (Cursor position 4).

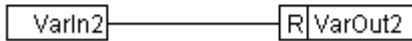
A negation can be deleted by negating again.

'Extras' 'Set/Reset' in CFC

Symbol:  **Shortcut:** <Ctrl> + <T>

This command can only be used for selected inputs of the element output .

The symbol for Set is S and for Reset is R.



VarOut1 is set to TRUE, if VarIn1 delivers TRUE. VarOut1 retains this value, even when VarIn1 springs back to FALSE.

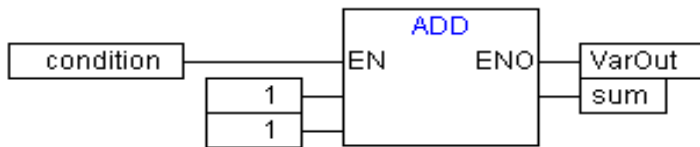
VarOut2 is set to FALSE, if VarIn2 delivers TRUE. VarOut2 retains this value, even when VarIn2 springs back to FALSE.

Multiple activation of this command causes the output to change between Set, Reset and the normal condition.

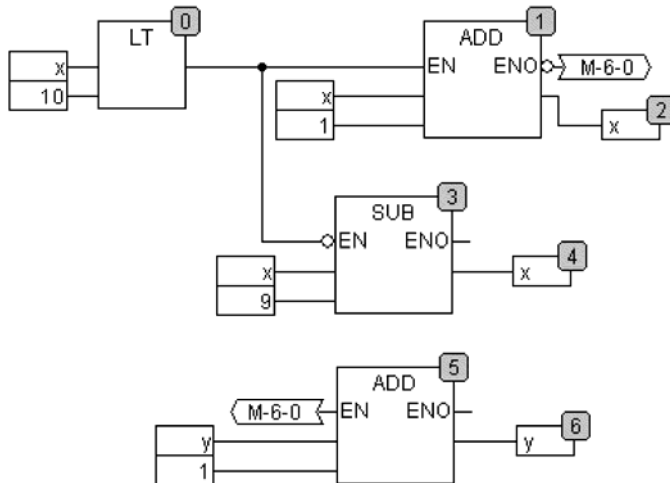
'Extras' 'EN/ENO' in CFC

Symbol:  **Shortcut:** <Ctrl> + <I>

This command is used to give a selected block (Cursor position 3) an additional Boolean enable input EN (Enable In) and a Boolean output ENO (Enable Out).



ADD is only executed in this example when the Boolean variable "condition" is TRUE. VarOut will also be set to TRUE after the execution of ADD. But if afterwards condition changes to FALSE, ADD will not be executed any more and thus VarOut remains TRUE! The example below shows how the value ENO can be used for further blocks:



x should initialised to 1 and y initialised to 0. The numbers in the right corner of the block indicate the order in which the commands are executed.

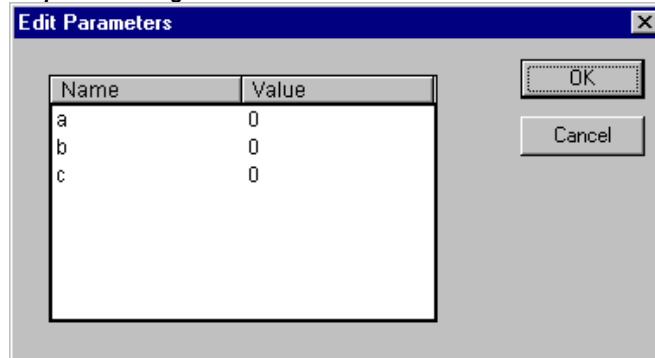
x will be increased by one until it reaches the value 10. This causes the output of the block LT(0) to deliver the value FALSE and SUB(3) and ADD(5) will be executed. x is set back to the value 1 and y is increased by 1. LT(0) is executed again as long as x is smaller than 10. y thus count the number of times x passes though the value range 1 to 10.

'Extras' 'Properties...' in CFC

Constant input parameters (VAR_INPUT CONSTANT) from functions and function blocks are not shown directly in the continuous function chart editor. These can be shown and their value can be

changed when one selects the trunk of the block in question and then selects the command 'Extras', 'Properties' or simply double clicks on the trunk. The „Edit parameters" dialog opens:

Properties dialog



The values of the constant input parameter (VAR_INPUT CONSTANT) can be changed. Here it is necessary to mark the parameter value in the column Value. Another mouse click or pressing on the space bar allows this to be edited. Confirmation of the change to the value is made by pressing the <Enter> key or pressing <Escape> rejects the changes. The button **OK** stores all of the changes which were made.

Please regard: This functionality and the associated declaration of variables with keyword "VAR_INPUT CONSTANT" is only of impact for the CFC editor. In the FBD editor always all INPUT variables will be displayed at a box, no matter whether declared as VAR_INPUT or VAR_INPUT CONSTANT. Also for text editors this does not make any difference.

Selecting elements in CFC

One clicks on the trunk of the element to select it.

To mark more elements one presses the <Shift> key and clicks in the elements required, one after the other, or one drags the mouse with the left hand mouse key depressed over the elements to be marked.

The command '**Extras**' '**Select all**' marks all elements at once.

Moving elements in CFC

One or more selected elements can be moved with the arrow keys as one is pressing on the <Shift> key. Another possibility is to move elements using a depressed left mouse key. These elements are placed by releasing the left mouse key in as far as they do not cover other elements or exceed the foreseen size of the editor. The marked element jumps back to its initial position in such cases and a warning tone sounds.

Copying elements in CFC

One or more selected elements can be copied with the command '**Edit**' '**Copy**' and inserted with the command '**Edit**' '**Paste**'.

Creating connections

An input of an element can be precisely connected to the output of another element. An output of an element can be connected to the inputs of a number of other elements.

There are a number of possibilities to connect the input of an element E2 with the output of an element E1.



Place the mouse on the output of element E1, click with the left mouse key, hold the left mouse key down and drag the mouse cursor onto the input of element E2 and let the left mouse key go. A connection is made from the output of element E1 to the mouse cursor during this dragging operation with the mouse.

Place the mouse on the input of element E2, click with the left mouse key, hold the left mouse key down and drag the mouse cursor onto the output of element E1 and let the left mouse key go.

Move one of the elements E1 or E2 and place it in such a way by letting go of the left mouse key that the output of element E2 and the input of element E1 touch.

Where element E2 is a block with a free input, a connection can also be made by dragging the mouse from an output from E1 to the trunk of E2. A connection with the free input at the highest position on E2 will be created when the mouse key is released. In the case where block E2 does not have a free input but is an operator which can have an input added to it, a new input will be automatically generated.

The output and input of a block can be connected together (feedback path) by using this method. To establish a connection between two pins, click with the left mouse button on one pin, hold the button down and thus drag the connection to the desired pin, where you then release the button. If during the dragging of the connection extends outside working area of the editor, scrolling occurs automatically. For simple data types, type testing is carried out during the connection. If the types of the two pins are not compatible, the cursor changes to „Forbidden“. For complex data types, no testing takes place.

Changing connections

A connection between the output of an element E1 and the input of an element E2 can easily be changed into a connection between the output of element E1 and the input of element E3. The mouse is clicked on the input of E2, the left mouse key is kept depressed, the mouse cursor is moved to the input of E3 and then released.

Deleting connections

There are a number of possibilities for removing the connection between the output of an element E1 and the input of an element E2:

Select the output of element E1 and press the <Delete> key or execute the command 'Edit' 'Delete'. Several connections will be removed at the same if the output of E1 is connected to more than one of inputs.

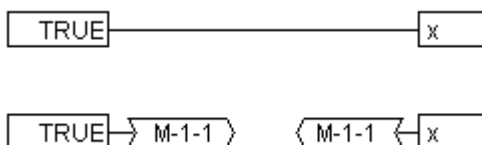
Select the input of element E2 and press the <Delete> key or execute the command 'Edit' 'Delete'.

Select the input of E2 with the mouse, hold the left mouse key depressed and drag the connection from the input to E2 away. The connection is removed when the left mouse key is released in a free area of the screen.

'Extras' 'Connection marker'

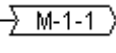
Connections can also be represented by a connector (connection marker) instead of a connecting line. Here the output and the associated input have a connector added to them which is given a unique name.

Where a connection already exists between the two elements which should now be represented by connectors, the output of the connecting line is marked and the menu point 'Extras' 'Connection marker' is selected. The following diagram shows a connection before and after the selection of this menu point.

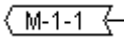


A unique name is given as standard by the program which begins with M, but which can be changed. The connector name is stored as an output parameter, but can be edited both at the input and at the output.

It is important to know that the connector name is associated with a property of the output of a connection and is stored with it.

1. Edit the connector at the output: 

If the text in the connector is replaced, the new connector name is adopted by all associated connectors at the inputs. One cannot, however, select a name which already belongs to **another** connection marker since the uniqueness of the connector name would be violated.

2. Edit the connector at the input: 

If the text in a connector is replaced, it will also be replaced in the corresponding connection marker on the other POU. Connections in connector representations can be converted to normal connections in that one marks the output of the connections (Cursor position 4) and again selects the menu point **'Extras' 'Connection marker'**.

Insert inputs/outputs "on the fly"

If exactly one input or output pin of an element is selected, then the corresponding input- or output-element can be directly inserted and its editor field filled with a string by entering the string at the keyboard.

Order of execution

The elements block, output, jump, return and label each possess a number indicating the order in which they are executed. In this sequential order the individual elements are evaluated at run time.

When pasting in an element the number is automatically given according to the topological sequence (from left to right and from above to below). The new element receives the number of its topological successor if the sequence has already been changed and all higher numbers are increased by one.

The number of an element remains constant when it is moved.

The sequence influences the result and must be changed in certain cases.

If the sequence is displayed, the corresponding sequential execution number is shown in the upper right hand corner of the element.

'Extras' 'Order' 'Show Order'

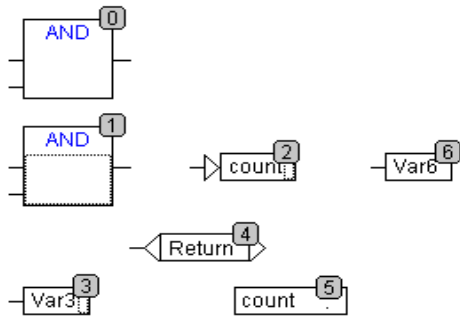
This command switches the display of the order of execution on and off. The default setting is to show it (recognised by a tick (✓) in front of the menu point).

The relevant order of execution number appears in the upper right hand corner for the elements block, output, jump, return and label.

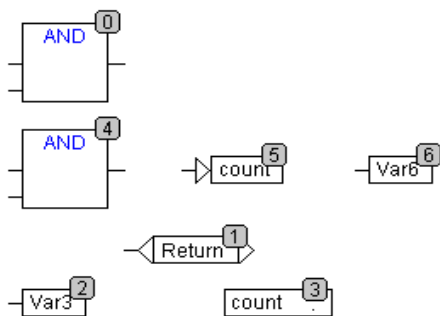
'Extras' 'Order' 'Order topologically'

Elements are ordered in a topological sequence when the execution takes place from left to right and from above to below, that is the number increases from left to right and from above to below for topologically arranged elements. The connections are not relevant, only the location of the elements is important.

All **selected** elements are topologically arranged when the command **'Extras' 'Order' 'Order topologically'** is executed. All elements in the selection are taken out of the sequential processing list by this process. The elements are then entered into the remaining sequential processing list individually from bottom right through to upper left. Each marked element is entered into the sequential processing list before its topological successor, i.e. it is inserted before the element that in a topological sequencing would be executed after it, when all elements in the editor were sequenced according to a topological sequencing system. This will be clarified by an example:



The elements with numbers 1, 2 and 3 are selected. If the command **'Order topologically'** is selected the elements are first taken out of the sequential processing list. Var3, the jump and the AND-operator are then inserted again one after the other. Var3 is placed before the label and receives the number 2. The jump is then ordered and receives the number 4 at first but this then becomes 5 after the AND is inserted. The new order of execution which arises is:



When a newly generated block is introduced it will be placed by default in front of its topological successor in the sequential processing list.

'Extras' 'Order' 'One up'

With this command all selected elements with the exception of the element which is at the beginning of the sequential processing list are moved one place forwards in the sequential processing list.

'Extras' 'Order' 'One down'

With this command all selected elements with the exception of the element which is at the end of the sequential processing list are moved one place backwards in the sequential processing list.

'Extras' 'Order' 'Order first'

With this command all selected elements will be moved to the front of the sequential processing list whereby the order within the group of selected elements is maintained. The order within the group of unselected elements also remains the same.

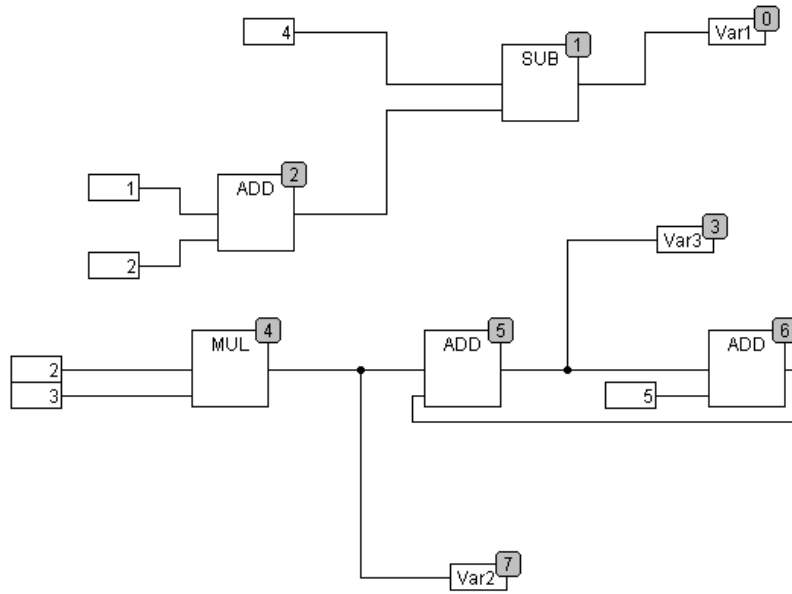
'Extras' 'Order' 'Order last'

With this command all selected elements will be moved to the end of the sequential processing list whereby the order within the group of selected elements is maintained. The order within the group of unselected elements also remains the same.

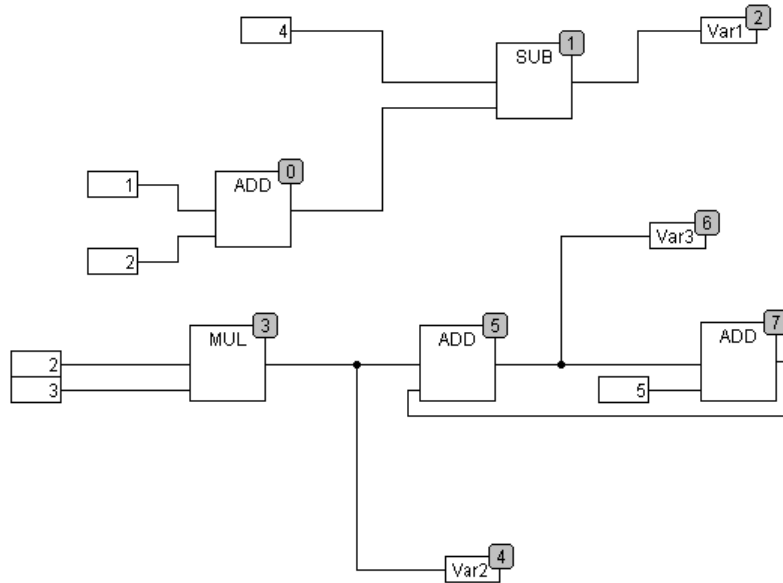
'Extras' 'Order' 'Order everything according to data flow'

This command **effects all** elements. The order of execution is determined by the data flow of the elements and not by their position.

The diagram below shows elements which have been ordered topographically.

Sequence before the ordering according to data flow

The following arrangement exists after selecting the command:

Sequence after the ordering according to data flow

When this command is selected the first thing to happen is that the elements are ordered topographically. A new sequential processing list is then created. Based on the known values of the inputs, the computer calculates which of the as yet not numbered elements can be processed next. In the above "network" the block AND, for example, could be processed immediately since the values at its inputs (1 and 2) are known. Block SUB can only then be processed since the result from ADD must be known first, etc.

Feedback paths are inserted last.

The advantage of the data flow sequencing is that an output box which is connected to the output of a block comes immediately after it in the data flow sequencing system which by topological ordering would not always be the case. The topological ordering can deliver another result in some cases than ordering by data flow, a point which one can recognise from the above example.

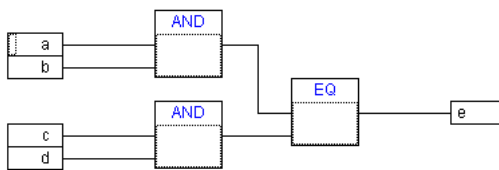
'Extras' 'Create macro'

Symbol:

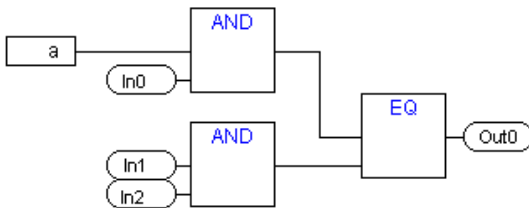
With this command, several POU's that are selected at the same time can be assembled into a block, which can be named as a macro. Macros only can be reproduced by Copy/Paste, whereby each copy becomes a separate macro whose name can be chosen independently. Macros are thus not references. All connections that are cut by the creation of a macro generate in- or out-pins on the macro. Connections to inputs generate an in-pin. The default name appears next to the pin in the form In<n>. For connections to outputs, Out<n> appears. Affected connections which had connection markers prior to the creation of the macro, retain the connection marker on the PIN of the macro.

At first, a macro has the default name "MACRO". This can be changed in the Name field of the macro use. If the macro is edited, the name of the macro will be displayed in the title bar of the editor window appended to the POU name.

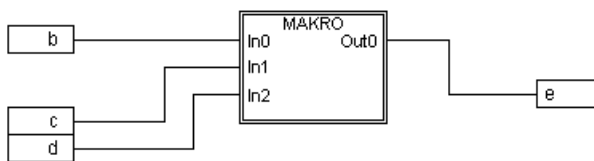
Example: Selection



Macro:



In the editor:



'Extras' 'Jump into Macro'

Symbol:

By this command, or by double clicking on the body of the macro, the macro is opened for editing in the editor window of the associated POU. The name of the macro is displayed appended to the POU name in the title bar.

The pin boxes generated for the in- and outputs of the macro during creation can be handled like normal POU in- and outputs. They can also be moved, deleted, added, etc. They differ only in how they are displayed and have no position index. For adding you can use the buttons (input) resp. (output), which are available in the menu bar. Pin boxes have rounded corners. The text in the pin-box matches the name of the pin in the macro display.

The order of the pins in the macro box follows the order of execution of the elements of the macro. A lower order index before a higher one, higher pin before lower.

The processing order within the macro is closed, in other words the macro is processed as a block, at the position of the macro in the primary POU. Commands for manipulating the order of execution therefore operate only within the macro.

'Extras' 'Expand macro'

With this command, the selected macro is re-expanded and the elements contained in it are inserted in the POU at the macro's location. The connections to the pins of the macro are again displayed as connections to the in- or outputs of the elements. If the expansion of the macro can not occur at the location of the macro box for lack of space, the macro is displaced to the right and down until enough space is available.

Note: If the project is saved under project version number 2.1, the macros will likewise all be expanded. All macros will also be expanded before conversion into other languages.

'Extras' 'One macro level back', 'Extras' 'All macro levels back'

Symbols: 

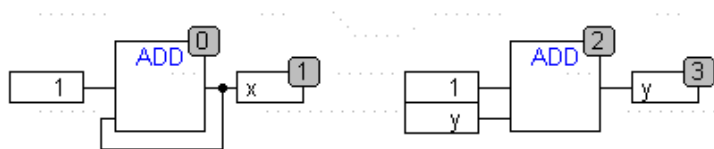
These commands are also available in the toolbar, as soon as a macro is opened for editing. If macros are nested within one another, it is possible to switch to the next higher or to the highest display level.

Feedback paths in CFC

Feedback paths can only be displayed directly in the continuous function chart editor and not in the usual function block diagram editor. Here it should be observed that the output of a block always carries an internal intermediate variable. The data type of the intermediate variable results, for operators, from the largest data type of the inputs.

The data type of a constant is obtained from the smallest possible data type, that is the constant '1' adopts the data type SINT. If now an addition with feedback and the constant '1' is executed, the first input gives the data type SINT and the second is undefined because of the feedback. Thus the intermediate variable is also of the type SINT. The value of the intermediate variable is only then allocated to the output variable.

The diagram below shows an addition with feedback and an addition with a variable. The variables x and y should be of the type INT here.



There are differences between the two additions:

The variable y can be initialised with a value which is not equal to zero but this is not the case for intermediate variable for the left addition.

The intermediate variable for the left addition has the data type SINT while that on the right has the data type INT. The variables x and y have different values after the 129th call up. The variable x, although it is of the type INT, contains the value 127 because the intermediate variable has gone into overflow. The variable y contains the value 129, on the other hand.

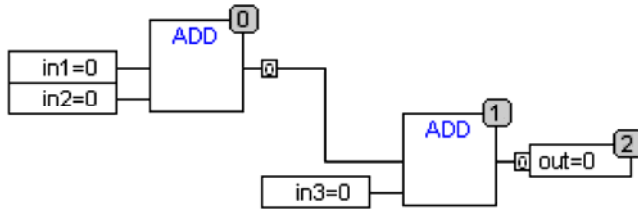
CFC in Online mode

Monitoring:

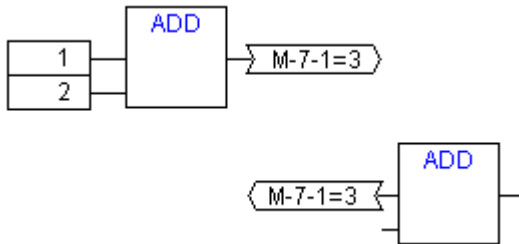
The values for inputs and outputs are displayed within the input or output boxes. Constants are not monitored. For non-boolean variables, the boxes are expanded to accommodate the values displayed.

For boolean connections, the variable name as well as the connection are displayed in blue if the value is TRUE, otherwise they remain black.

Internal boolean connections are also displayed Online in blue in the TRUE state, otherwise black. The value of internal non-boolean connections is displayed in a small box with rounded corners on the output pin of the connection.



PINs in macros are monitored like in- or output boxes.



Non-boolean connections with connection markers display their value within the connection marker. For boolean connections, the lines as well as the marker names are displayed in blue if the line is carrying the value TRUE, otherwise black.

Flow control:

When flow control is switched on, the connections that have been traversed are marked with the color selected in the project options.

Breakpoints:

Breakpoint s can be set on all elements that also have a processing sequence order index. The processing of the program will be halted prior to execution of the respective element that is for POU's and outputs before the assignment of inputs, for jump labels before execution of the element with the next index. The processing sequence index of the element is used as the breakpoint position in the Breakpoint dialog.

The setting of breakpoints on a selected element is accomplished with the F9 key or via the menu item 'Breakpoint on/off' in the 'Online' or 'Extras' menu or in the editor's context menu. If a breakpoint is set on an element, then this will be erased and reversed the next time the command 'Breakpoint on/off' is executed. In addition, the breakpoint on an element can be toggled by double-clicking on it.

Breakpoints are displayed in the colors entered in the project options.

RETURN label:

In Online mode, a jump label with the name „RETURN" is automatically generated in the first column and after the last element in the editor. This label marks the end of the POU and is jumped to when stepping just before execution leaves the POU. No RETURN marks are inserted in macros.

Stepping:

When using 'Step over' the element with the next-higher order index will always be jumped to. If the current element is a macro or a POU, then its implement branches when 'Step in' is in effect. If a 'Step over' is executed from there, the element whose order index follows that of the macro is jumped to.

Zoom to POU

Shortcut: <Alt>+<Enter>

With this command a selected POU is loaded into its editor. The command is available in the context menu (<F2>) or in the 'Extras' menu, if the cursor is positioned on the name of a POU in a text editor or if the POU box is selected in a graphic editor.

If you are dealing with a POU from a library, then the library manager is called up, and the corresponding POU is displayed.

The Graphic Editors

6 The Ressources

6.1 Overview of the Ressources

In the **Resources** register card of the Object Organizer, there are objects for configuring and organizing your project and for keeping track of the values of the variables:

Global Variables that can be utilized in the entire project; the global variables of the project as well as the libraries.

Alarm configuration for the organization of an alarm system in the CoDeSys project.

Library Manager for handling all libraries which are included to the project.

Log for recording the activities during the online sessions.

PLC Configuration for configuring your hardware.

Watch and Receipt Manager for indicating and presetting variable values.

Task Configuration for controlling your program control via tasks.

Target settings for selecting the hardware platform (target) and if available for customizing target specific parameters.

Workspace as an image of the project options.

Depending on the target settings the following resources also might be available:

Parameter Manager for managing variables, which are also accessible for other participants in the network. This functionality will only be available if defined in the corresponding target settings.

PLC Browser for monitoring of information from the PLC.

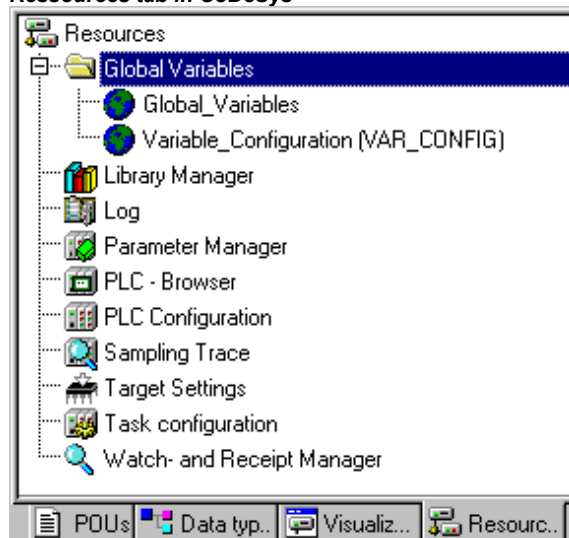
Sampling Trace for graphic logging of variable values.

Tools for linking external tools, which then can be started in CoDeSys. This functionality will only be available if defined in the corresponding target settings.

The **SoftMotion** functionality (license needed): **CNC program list** (CNC Editor) and **CAMs** (CAM-Editor) (see the separate documentation on SoftMotion).

Additionally there might be created and loaded a **Docuframe** file which offers a set of comments for the project variables (e.g. in a certain language), which will be printed when documenting the project with 'Project' 'Document'.

Ressources tab in CoDeSys



6.2 Global Variables, Variable Configuration, Document Frame

Objects in 'Global Variables'

In the Object Organizer, you will find three objects in the **Resources** register card in the **Global Variables** folder (default names of the objects in parentheses).

Global Variables List (Global Variables)

Variables Configuration (Variable Configuration)

All variables defined in these objects are recognized throughout the project.


If the global variables folder is not opened (plus sign in front of the folder), you can open it with a double-click<Enter> in the line.

Select the corresponding object. The **'Object Open'** command opens a window with the previously defined global variables. The editor for this works the same way as the declaration editor.

Several Variables Lists

Global variables, global network variables (**VAR_GLOBAL**), global network variables (**VAR_GLOBAL**, target specific) and variable configurations (**VAR_CONFIG**) must be defined in separate objects.

If you have declared a large number of global variables, and you would like to structure your global variables list better, then you can create further variables lists.

In the Object Organizer, select the **Global Variables** folder or one of the existing  objects with global variables. Then execute the **'Project' 'Object Add'** command. Give the object that appears in the dialog box a corresponding name. With this name an additional object will be created with the key word **VAR_GLOBAL**. If you prefer an object a variable configuration, change the corresponding key word to **VAR_CONFIG**.

6.2.1 Global Variables

What are Global Variables

„Normal“ variables, constants or remanent variables that are known throughout the project can be declared as global variables, but also network variables that are also used for data exchange with other network subscribers.

Please regard: In a project you can define a local variable which has the same name like a global variable. In this case within a POU the locally defined variable will be used.

It is not allowed to name two global variables identically. For example you will get a compiler error, if you have defined a variable "var1" in the PLC Configuration as well as in a global variables list.

Network variables

Note: The use of network variables must be supported by the target system and must be activated in the target settings (category Network functionality).

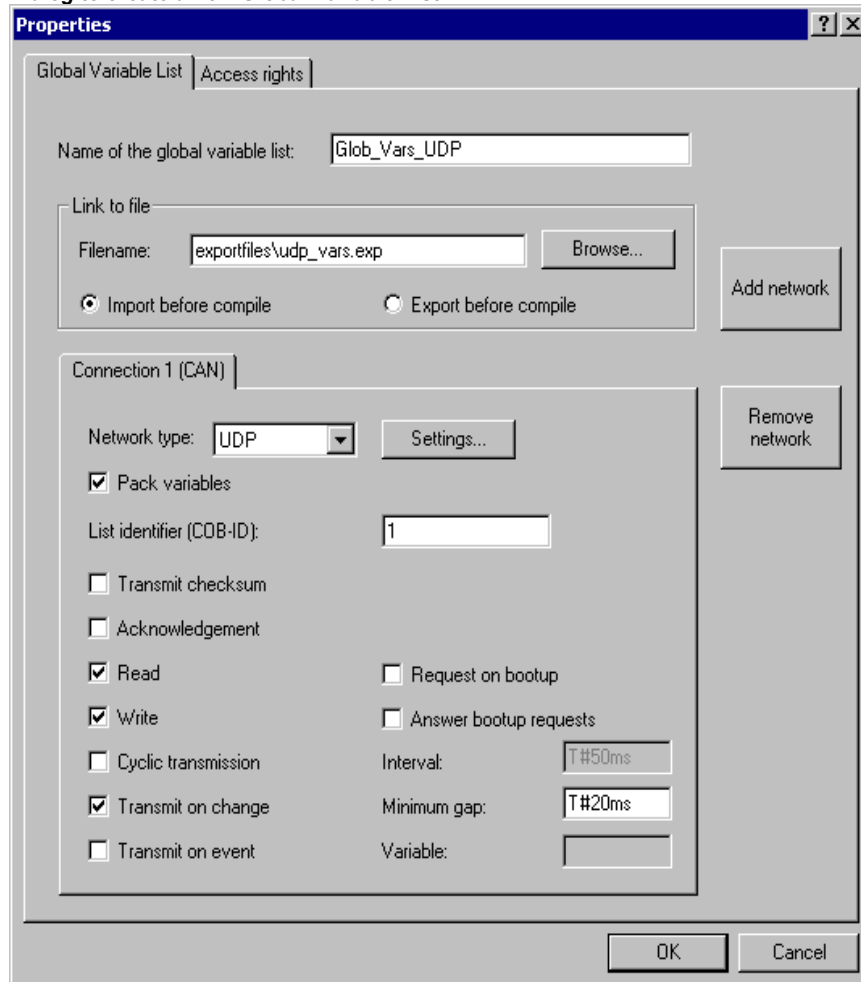
By an automatic data exchange (compare this to the non-automatic data exchange via the Parameter Manager) it is possible to update the value of a network variable on several controller systems within a **CoDeSys** compatible controller network. This requires no controller-specific functions but the network subscribers must use identical declaration lists and matching transfer configurations for network variables in their projects. In order to make this possible it is recommended that the declaration not be entered manually in each controller application, but loaded from a separate file when creating the list. (see 'Create a global variables list').

Create a Global Variable List

To create a Global Variable List, open the register 'Resources' in the Object Organizer and select the entry 'Global Variables' or select an already existing list. Then choose the command 'Project' 'Object' 'Add' to open the dialog Global variable list.

This dialog can also be opened by the command 'Project' 'Object' 'Properties' which is available if an existing Global Variable List is marked in the object organizer. It shows the configuration of this list..

Dialog to create a new Global Variable List




Name of the global variable list: Insert a list name.

Link to file:

Filename: If you have an export file (*.exp) or a DCF file, which contains the desired variables, you can set up a link to this file. To do this, insert the path of the file in the field **Filename** resp. press the button **Browse** to get the standard dialog 'Select text file'. DCF files are converted to ICE syntax when they are read in.

Activate option **Import before compile**, if you wish that the variable list will be read from the external file before each compilation of the project. Activate the option **Export before compile**, if you want the variable list to be written to the external file before each compilation of the project.

If you close the 'Global variable list' dialog with **OK**, the new object is created. Global variables lists can be recognized in the Object Organizer by the symbol . With the command 'Project' 'Object' 'Properties' you can re-open the 'Global variable list' configuration dialog for the entry marked in the Object Organizer.

Configuration of network variables:

If the option 'Support network variables' is activated in the target settings, then the button <Add network> is available. Pressing this button the dialog gets extended and looks like shown in the picture. If the option is not activated, the button is not available.

Connection <n> (<Network type>): In the lower part of the dialog you can create configuration sets for up to four network connections, each on a separate tab. A configuration set defines the parameters of the data exchange for the particular variables list within the network. In order for the exchange in the network to work as intended, the same variable list must be compatibly configured to match in the other network subscribers.

If no configuration is yet present, you will get in the case of a UDP network a single tabulator sheet with the inscription '**Connection 1 (UDP)**'. Each time the 'Add network' button is pressed again, you get up to four more sheets inscribed with serial numbers after „Connection“.

Network type: Choose the desired type from the list. The list is defined by the target system entries. For example, „CAN“ as an abbreviation for a CAN network, or „UDP“ for a UDP transmission system, might be selectable.

Settings: This button opens the dialog **Settings for <networktype>** with the following configuration parameters:

UDP:

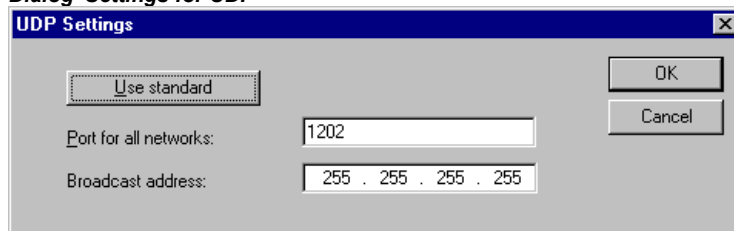
Use standard If this button is pressed, Port 1202 will be defined for the data exchange with the other network participants. The Broadcast/Multicast address will be set to "255 . 255 . 255 . 255" , which means, that the data exchange will be done with all participants in the network.

Port: Enter here a desired port number to overwrite the standard setting (see above, Use standard). Make sure that the other nodes in the network define the same port! If you have more than one UDP connection defined in the project then the port number will be automatically modified in all configuration sets according to the input you make here.

Broadcast/Multicast address: Enter here an address resp. the address range of a sub-network, if you want to overwrite the standard setting (e.g. "197 . 200 . 100 . 255", if you want to communicate with all nodes with IP-addresses 197 . 200 . 100 . x).

Regard for Win32 systems, that the Broadcast addresses must match the subnet mask of the TCP/IP configuration of the PC !

Dialog 'Settings for UDP



CAN:

Controller Index: Index of the CAN Controller, by which the variables should be transferred.

The following options can be activated or deactivated in configuring the transmission behaviour of the variables:

Pack variables: The variables are assembled for transfer into packets (telegrams) whose size depends on the network. If the option is deactivated, a packet is set up for each variable.

List identifier (COB-ID): Identification number of the first packet, in which the variables will be sent. (default = 1). Further packets will be numbered in ascendant order. Note: A COB-ID **must be unique** within all network variables lists used in the project. Using the same ID in multiple list configurations will cause a compile error with compiler versions >= 2.3.7.0.

It depends on the target system, whether the network variables of the list can be defined to be 'readable' and 'writing' or exclusively one of both. To set this property activate the respective options 'Read' and 'Write':

Read: The variables in the list will be read; if the option is deactivated, further variables sent over the net will be ignored. The following option can be activated in addition:

Request at Bootup: If the local node is a "reading" node (Option 'Read' activated), then as soon as it gets re-booted the actual variable values will be requested from all writing nodes and will be sent by those, independently of any other transmit conditions (time, event), which normally trigger the communication. Precondition: In the configuration of the writing nodes the option 'Answer Bootup requests' must be activated ! (see below).

Write: The variables will be written, the following options can be set additionally:

Include Checksum: A checksum will be added to each packet which is sent. The checksum will be checked by the receiver to make sure that the variable definitions of sender and receiver are identical. A packet with a non-matching checksum will not be accepted and – if this is configured ('Use acknowledge transfer', see below) – will be acknowledged negatively.


Use acknowledged transfer: (no function in case of CAN) Each message will be acknowledged by the receiver. As soon as the sender does not get at least one acknowledgement within a cycle, an error message will be produced, which in case of an UDP-network will be written to the diagnosis structure provided by NetVarUdp_LIB_V23.lib.

Answer Bootup requests: If the local node is a "writing" node (Option 'Write' activated), then each request of a reading node which is sent by it at bootup (Option Request on Bootup, see above), will be answered. That means that the actual variable values will be transmitted even if none of the other defined transmission triggers (time or event) would force this at this moment.

Transmit each cycle: Variables are written within the intervals specified after **Interval**. (time notation e.g. T#70ms).

Transmit on change: Variables are written only when their values change; an entry after Minimum can, however, set a minimum time lapse between transfers.

Transmit on event: The variables of the list will be written as soon as the variable inserted at **Variable** gets TRUE.

Global Network variables lists are marked by the symbol  in the Object Organizer.

Note:	If a network global variable is used on one or more tasks , the following applies to the time component of the transfer: When each task is called it is tested to determine which parameters apply to the transfer of the variable value (configuration in the 'Global variables list' dialog). The variable value will be transferred or not, depending on whether the specified time interval has passed. At each transfer the time interval counter for this variable is reset to zero.
--------------	---

Sending is always undertaken from the run-time system of the controller affected. Thus no control-specific functions have to be provided for the data exchange.

Editing Global Variable and Network Variable Lists

The editor for global variables works similar to the declaration editor. But note that you cannot edit in this editor an list, which is an image of an linked external variable list ! External variable lists only can be edited externally and they will be read at each opening and compiling of the project.

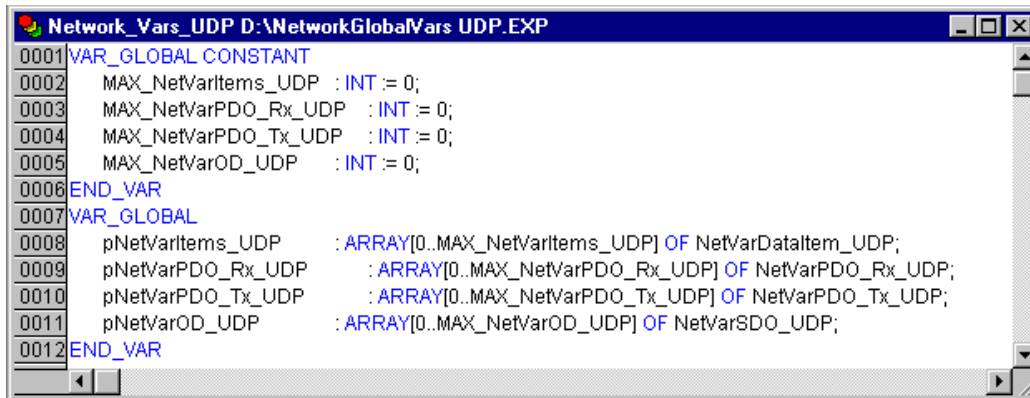
Syntax:

```
VAR_GLOBAL
(* Variables declarations *)
END_VAR
```

Network variables only can be used, if allowed by the target system. They are also defined in this syntax.

Example of a network variables list which was created by linking of an export file *.exp and which got the name NETWORKVARIABLES_UDP:

Example of a network variables list, which has been created by loading an export file *.exp and which was named Network_Vars_UDP:



```

0001 VAR_GLOBAL CONSTANT
0002   MAX_NetVarItems_UDP :INT := 0;
0003   MAX_NetVarPDO_Rx_UDP :INT := 0;
0004   MAX_NetVarPDO_Tx_UDP :INT := 0;
0005   MAX_NetVarOD_UDP :INT := 0;
0006 END_VAR
0007 VAR_GLOBAL
0008   pNetVarItems_UDP :ARRAY[0..MAX_NetVarItems_UDP] OF NetVarDataItem_UDP;
0009   pNetVarPDO_Rx_UDP :ARRAY[0..MAX_NetVarPDO_Rx_UDP] OF NetVarPDO_Rx_UDP;
0010   pNetVarPDO_Tx_UDP :ARRAY[0..MAX_NetVarPDO_Tx_UDP] OF NetVarPDO_Tx_UDP;
0011   pNetVarOD_UDP :ARRAY[0..MAX_NetVarOD_UDP] OF NetVarSDO_UDP;
0012 END_VAR

```

Editing Remanent Global Variables Lists

If they are supported by the runtime system, remanent variables may be processed. There are two types of remanent global variables (see also Chapter 5.2.1, Remanent Variables !):

Retain variables remain unchanged after an uncontrolled shutdown of the runtime system (off/on) or an 'Online' 'Reset' in CoDeSys. **Persistent variables** remain unchanged only after a download.

Persistent variables are not automatically also Retain variables !

Remanent variables are additionally assigned the keyword **RETAIN** or **PERSISTENT** or both.

See chapter 5.2.1, Remanent Variables, for further information.

Network variables are also defined in this syntax.

Syntax:

```

VAR_GLOBAL RETAIN
(*_Variables declarations *)
END_VAR

VAR_GLOBAL PERSISTENT
(*_Variables declarations *)
END_VAR

```

For the combination of retain and persistent properties both keywords are used:

```

VAR_GLOBAL RETAIN PERSISTENT OF VAR_GLOBAL PERSISTENT RETAIN

```

Network variables (target specific) are also defined using this syntax.

Global Constants

Global constants additionally get the keyword **CONSTANT**.

Syntax:

```

VAR_GLOBAL CONSTANT
(*_Variables declarations *)
END_VAR

```

6.2.2 Variable Configuration

In function blocks it is possible to specify addresses for inputs and outputs that are not completely defined, if you put the variable definitions between the key words **VAR** and **END_VAR**. Addresses not completely defined are identified with an asterisk.

Example:

```

FUNCTION_BLOCK locio
VAR
  loci AT %I*: BOOL := TRUE;

```

```

    loco AT %Q*: BOOL;
END_VAR

```

Here two local I/O-variables are defined, a local-In (%I*) and a local-Out (%Q*).

If you want to configure local I/Os for variables configuration in the Object Organizer in the **Resources** register card, the object **Variable_Configuration** will generally be available. The object then can be renamed and other objects can be created for the variables configuration.

The editor for variables configuration works like the declaration editor.

Variables for local I/O-configurations must be located between the key words **VAR_CONFIG** and **END_VAR**.

The name of such a variable consists of a complete instance path through which the individual POU's and instance names are separated from one another by periods. The declaration must contain an address whose class (input/output) corresponds to that of the incompletely specified address (%I*, %Q*) in the function block. Also the data type must agree with the declaration in the function block.

Configuration variables, whose instance path is invalid because the instance does not exist, are also denoted as errors. On the other hand, an error is also reported if no configuration exists for an instance variable. In order to receive a list of all necessary configuration variables, the "All Instance Paths" menu item in the 'Insert' menu can be used.

Example for a Variable Configuration

Assume that the following definition for a function block is given in a program:

```

PROGRAM PLC_PRG
VAR
Hugo: locio;
Otto: locio;
END_VAR

```

Then a corrected variable configuration would look this way:

```

VAR_CONFIG
PLC_PRG.Hugo.loci AT %IX1.0 : BOOL;
PLC_PRG.Hugo.loco AT %QX0.0 : BOOL;
PLC_PRG.Otto.loci AT %IX1.0 : BOOL;
PLC_PRG.Otto.loco AT %QX0.3 : BOOL;
END_VAR

```

'Insert' 'All Instance Paths'

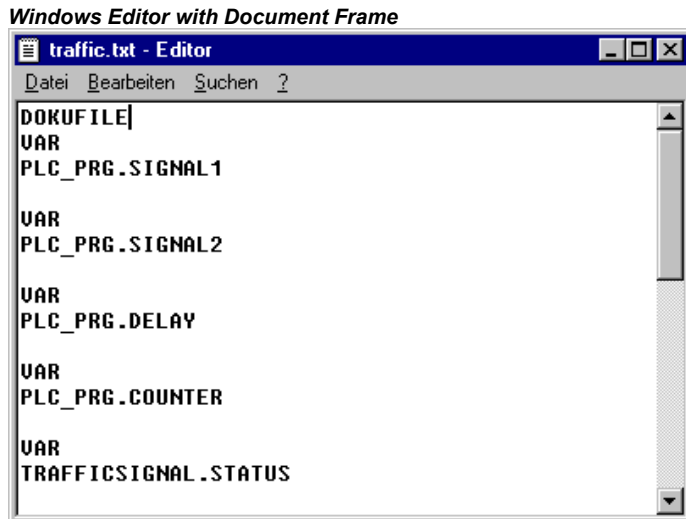
With this command a **VAR_CONFIG - END_VAR**-block is generated that contains all of the instance paths available in the project. Declarations already on hand do not need to be reinserted in order to contain addresses already in existence. This menu item can be found in the window for configuration of variables if the project is compiled ('Project' 'Rebuild All').

6.2.3 Document Frame

If a project is to receive multiple documentations, perhaps with German and English comments or if you want to document several similar projects that use the same variable names, then you can save yourself a lot of work by creating a docuframe with the 'Extras' 'Make Docuframe File' command.

The created file can be loaded into a desired text editor and can be edited. The file begins with the **DOCUFILE** line. Then a listing of the project variables follows in an arrangement that assigns three lines to each variable: a **VAR** line that shows when a new variable comes; next, a line with the name of the variable; and, finally, an empty line. You can now replace this line by using a comment to the variable. You can simply delete any variables that you are unable to document. If you want, you can create several document frames for your project.

In order to use a document frame, give the 'Extras' 'Link Docu File' command. Now if you document the entire project, or print parts of your project, then in the program text, there will be an insertion of the comment produced in the docuframe into all of the variables. This comment only appears in the printout!



'Extras' 'Make Docuframe File'

Use this command to create a document frame. The command is at your disposal, whenever you select an object from the global variables.

A dialog box will open for saving files under a new name. In the field for the **name file**, the *.txt extension has already been entered. Select a desired name. Now a text file has been created in which all the variables of your project are listed.

'Extras' 'Link Docu File'

With this command you can select a document frame.

The dialog box for opening files is opened. Choose the desired document frame and press **OK**. Now if you document the entire project, or print parts of your project, then in the program text there will be an insertion of the comment produced in the docuframe into all of the variables. This comment only appears in the printout!

To create a document frame, use the 'Extras' 'Make Docuframe File' command.

6.3 Alarm Configuration

6.3.1 Overview

The alarm system integrated in CoDeSys allows detecting critical process states, recording them and visualizing them for the user with the aid of a visualization element. The alarm handling can be done in CoDeSys or alternatively in the PLC. For alarm handling in the PLC please see the target settings category 'Visualization'.

If supported by the target system, the dialogs for the 'Alarm configuration' are available in the 'Resources' tab.

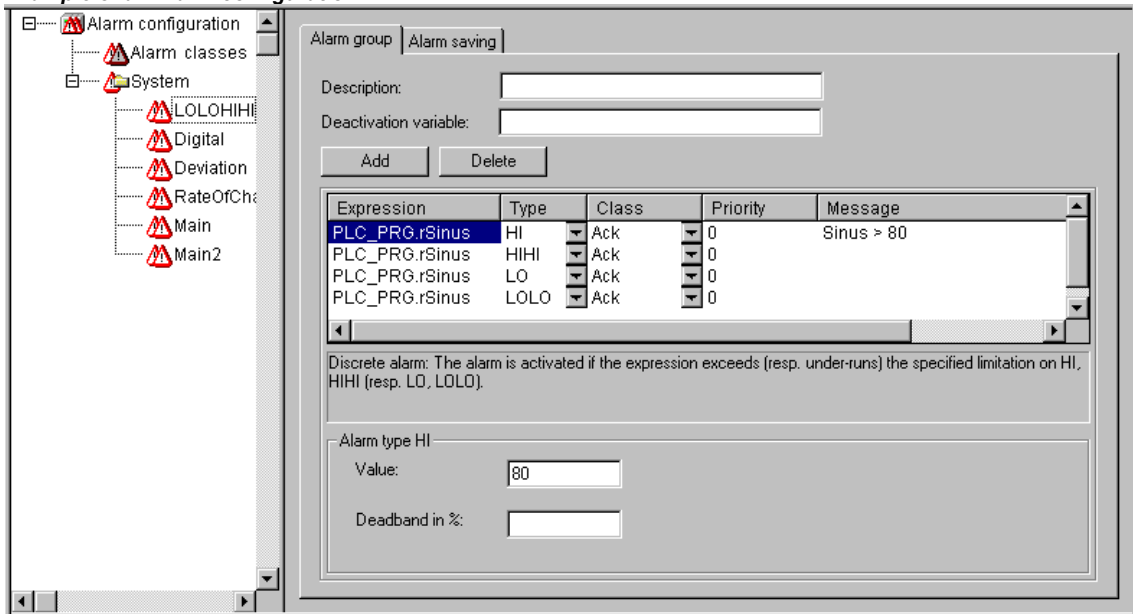
Here you define **Alarm classes** and **Alarm groups**. An alarm class serves for the typing of an alarm, that means it assigns certain parameters to the alarm. An alarm group serves for the concrete configuration of one or several alarms (which get assigned a certain class and further parameters) for the use in the project. Thus a class is useful for structuring the available alarms. The different alarm groups are defined by the user by inserting appropriate entries below the header **'System'** in the configuration tree.

For the **visualization** of alarms the element 'Alarm table' is available in the CoDeSys visualization. Using this table the user can watch and acknowledge alarms.

If a **History**, i.e. recording of Alarm-Events should be written to a log-file, such a file must be defined and for each alarm group the saving behaviour must be defined.

When you open the 'Alarm configuration' in the Resources tab, the **dialog 'Alarm configuration'** opens with a bi-partite window, which concerning the mode of operation is similar to that of the PLC Configuration or Task configuration. In the left part the configuration tree is displayed, in the right part the appropriate configuration dialog will be opened.

Example of an Alarm configuration



Open by a mouse-click on the plus sign at the entry 'Alarm configuration' the currently available configuration tree. If you are going to create a new configuration, this tree only will show the entries 'Alarm classes' and 'System'.

6.3.2 General information on alarms, Terms

The usage of an alarm system in CoDeSys obeys the following universal descriptions and definitions concerning alarms:

Alarm: Generally an alarm is regarded as a special condition (expression value).

Priority: The priority, also named "severity", of an alarm describes how important (severe) the alarm condition is. The highest priority is "0", the lowest valid priority value is "255".

Alarm state: An expression/variable configured for the alarm control can have the following states: NORM (no alarm), INTO (alarm just has come), ACK (alarm has come and has been acknowledged by the user), OUTOF (alarm state has been terminated, alarm "has gone", but not yet acknowledged!)

Sub-State: An alarm condition can have limits (Lo, Hi) and "extreme" limits (LoLo, HiHi). Example: The value of an expression ascends and first will transit the HI-limit, thus causing the coming of an HI-alarm. If the value continues ascending and exceeds also the HIHI-limit before the alarm gets acknowledged by the user, then the HI-alarm will get acknowledged automatically and just the HIHI-alarm remains in the alarm list (which is an internal list used for alarm administration). The HI-state in this case is named sub-state.

Acknowledgement of alarms: The main purpose of alarms is to inform the user on alarm situations. In doing so it often is necessary to make sure that the user has noticed this information (see possible actions assigned to an alarm in the alarm class configuration). The user must acknowledge the alarm in order to get the alarm removed from the alarm list.

Alarm Event: An alarm event must not be mixed up with an **alarm condition**. While an alarm condition can be valid for a longer period of time, an alarm event just describes the momentary

occurrence of a change, e.g. a change from the normal state to the alarm state. In the CoDeSys alarm configuration for the three types of events and the corresponding alarm states the same names are used (INTO, ACK, OUTOF).

In CoDeSys the following features are supported:

- Deactivation of the alarm generation for single alarms as well as for alarm groups
- Selection of the alarms which should be displayed by defining alarm groups and priorities
- Saving of all alarm events in an alarm table
- Visualization element 'Alarm table' in the CoDeSys Visualization

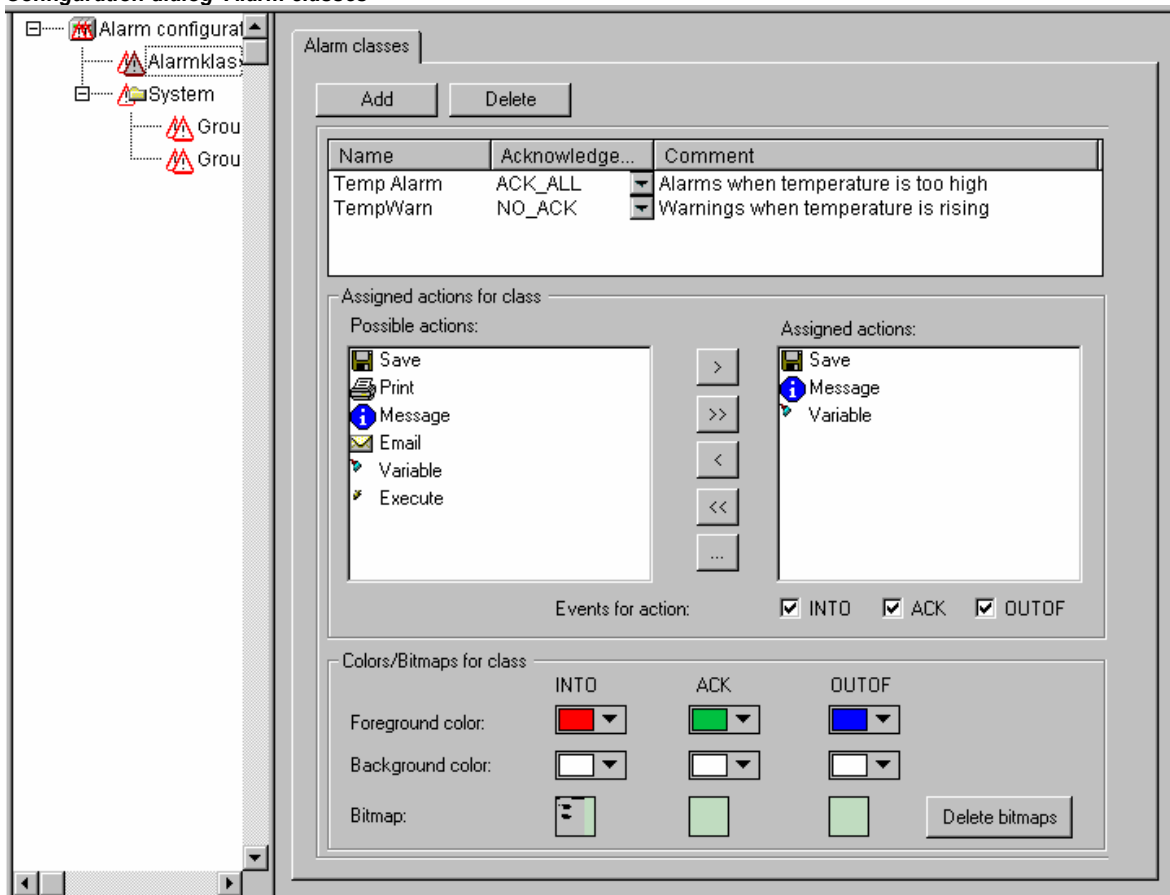
6.3.3 Alarm classes

Alarm classes are used for the general description of certain alarm criteria, such as how to handle acknowledgements (confirmation of an alarm by the user), which actions should automatically run as soon as a particular alarm state has been detected and which colors and bitmaps are to be used for a visualization of an Alarm table. Alarm classes are defined globally in the Alarm configuration and are then available as a base configuration when configuring alarm groups.

Configuration of alarm classes:

Select entry 'Alarm classes' in the alarm configuration tree. The configuration dialog 'Alarm classes' gets opened:

Configuration dialog 'Alarm classes'



Press button **Add** in order to create a new alarm class. Thereupon in the upper window a line will be inserted, primarily only with an entry "NOACK" (no acknowledgement) in the 'Acknowledgement' column. Define a name for the alarm class in the corresponding field in the **Name** column (open an edit frame by a mouse-click on the field) and if necessary modify the acknowledgement type in column **Acknowledgement**.

The following acknowledgements are available:

NO_ACK: No acknowledgement of the alarm by the user is required

ACK_INT0: A "come" alarm condition (status "INT0", alarm occurs) must be confirmed by the user.

ACK_OUTOF: A "gone alarm" (status "OUTOF", alarm terminated) must be confirmed by the user.

ACK_ALL: Gone and come alarm conditions must be confirmed by the user.

Additionally you can enter a **Comment**.

Entries for further alarm classes each will be added at the end of the list.

Use button **Delete** to remove the currently selected entry from the list.

Assigned actions for class <class name>:

Each alarm class defined in the upper window can get assigned a list of actions, which should be performed as soon as a Alarm Event occurs.

In the list of **Possible actions** select one and press button ">" to get it into the field **Assigned actions**. This field will finally contain the selection of actions assigned to the alarm class. Via button ">>" you can add all actions at a single blow. Via "<" resp.. "<<" you can remove one or all actions from the done existing selection. If an action is marked in the 'Assigned actions' list, via "..." a corresponding dialog can be opened to define the desired e-mail settings, the printer settings, the process variable resp. the executable program and, if applicable, a message text.

The following action types (**Possible actions**) are supported (for a definition of a message text see below):

Action	Description	Settings to be done in the corresponding dialog:
Save:	The alarm event will be saved internally, in order to be given out e.g.in a log-file. Please regard: In this case the log-file must be defined in the configuration of the alarm group !	The settings are done in the Alarm group definition in the Alarm saving dialog
Print:	A message text is sent to a printer.	Printer: Select one of the printers defined on the local system; Outputtext: Message text (see below) which should be printed out. Please regard: This function is not supported for Target-Visualization!
Message:	In the current visualization of the alarm a message window will be opened showing the defined text.	Message: Message text to be displayed in the message window. Please regard: This function is not supported for Target-Visualization!
E-Mail:	An e-mail containing the defined message will be sent.	From: e-mail address of sender; To: e-mail address of recipient; Subject: any subject; Message: Message text (see below); Server: Name of the e-mail server.
Variable:	A variable of the CoDeSys program will get the alarm status resp. a message text string.	Variable: Variable name: You can select project variables via the input assistant (<F2>): A boolean variable will indicate the alarm states NORM =0 and INTO=1, an integer variable will indicate the alarm states NORM =0, INTO =1, ACK =2, OUTOF =4; a string variable will get

		the message text defined in field; Message (see below)
Execute:	An executable file will be started as soon as the alarm event occurs.	Executable file: name of the file to be executed (e.g. notepad.exe, you can use the "..." button to get the standard dialog for selecting a file; Parameter: appropriate parameter(s) which should be attached to the call of the exe-file

Definition of the message text:

For action types 'Message', 'Print', 'Email' or 'Variable' you can define a message text which should be output in case of an Alarm Event.

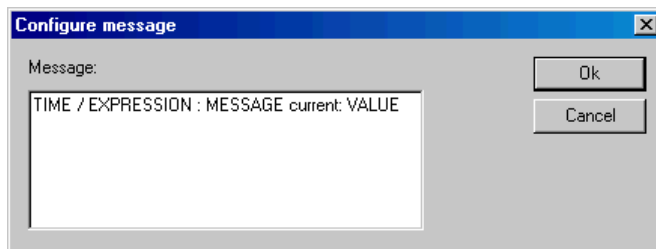
Line breaks at the text definitions in 'Message', 'Email' or 'Variable' can be inserted by <Ctrl>+<Enter>.

The following **placeholders** can be used when defining the alarm message:

MESSAGE	The message text which is defined for the particular alarm in the configuration of the alarm group will be used.
DATE	Date, when the alarm status was reached (INTO).
TIME	Time of alarm entry.
EXPRESSION	Expression (defined in alarm group) which has caused the alarm.
PRIORITY	Priority of the alarm (defined for alarm group.)
VALUE	Current value of the expression (see above).
TYPE	Alarm type (defined in alarm group)
CLASS	Alarm class (defined in alarm group)
TARGETVALUE	Target value for alarm types DEV+ and DEV- (defined in alarm group)
DEADBAND	Tolerance of the alarm (defined in alarm group)
ALLDEFAULT	Any information on the alarm will be output, like described for the line entries in a log file (History).

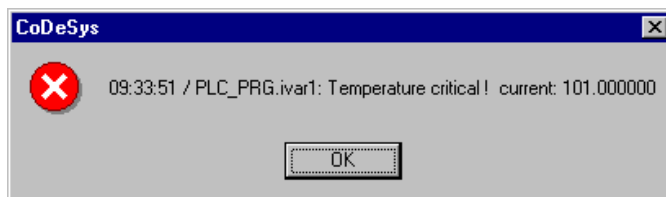
Example of defining an alarm message:

For a definition of a message box enter the following in the message window:



Further on when defining the alarm in the alarm group enter in column 'Message' the following:

"Temperature critical !". The output of the final alarm message will be like follows:



Note: The message text will also be affected in case of a **change of the project language** if it is included in a *.vis-file or a translation file *.tlt. BUT: In this case - like texts referring to a visualization it has to be set between two "#" -characters (e.g. in the example shown above : "#Temperature critical !#" and "TIME /EXPRESSION: MESSAGE #current#: VALUE", in order to get the text entered in the translation file as ALARMTEXT_ITEMS.)

A **log file** for action 'Save' is to be defined in the configuration of the alarm group (see Chapter 6.3.4).

Alarm Events for actions:

For each action you define, at which alarm events it should be started.

Activate the desired events:

INTO The alarm occurs. Status = INTO.

ACK Acknowledgement by the user has been done. Status = ACK.

OUTOF Alarm state terminated. Status = OUTOF.

Colors/Bitmaps for class <class name>

Each alarm class can get assigned own colors and bitmaps, which will be used for the differentiation of the alarms in the visualization element alarm table. Select a **Foreground** color and **Background color** for the possible events INTO, ACK and OUTOF (see above). The standard dialog for selecting a color will open as soon as you perform a mouse-click on the color symbol. For selecting a bitmap a mouse-click on the grey rectangle will open the standard dialog for selecting a file.

6.3.4 Alarm groups

Alarm groups are used for organizing the available alarms. Each alarm is definitely assigned to right one alarm group and is managed by this group. All alarms of a group can get assigned a common deactivation variable and common parameters regarding the alarm saving. Regard that even a single alarm must be configured within an alarm group.

A hierarchical structure of alarm groups can be defined via folder elements. When a alarm group is selected in the configuration tree, automatically the dialog **Alarm group** will be displayed.

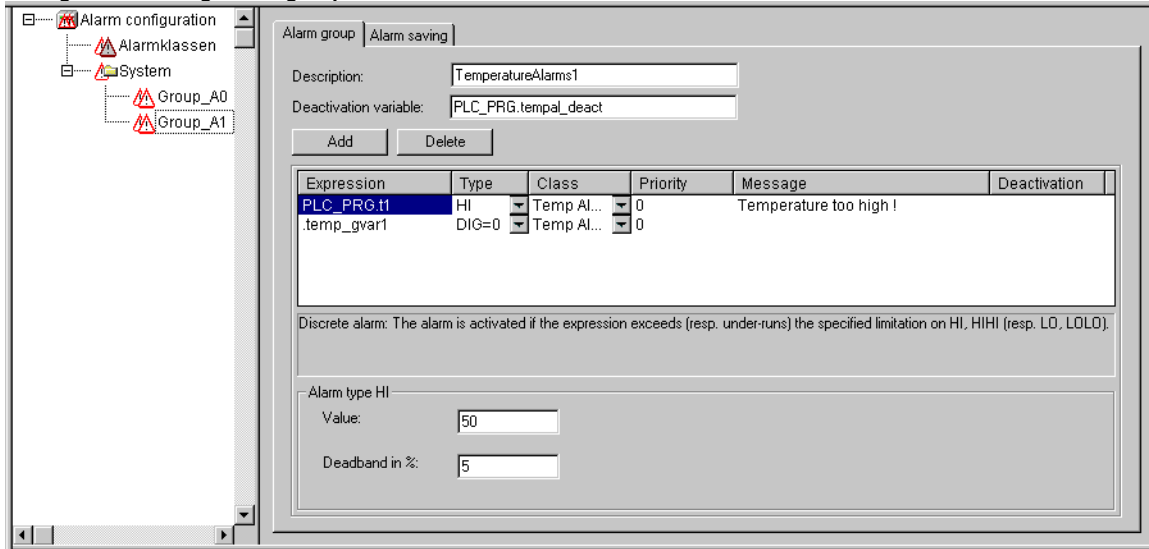
In the field **Description** enter a name for the alarm group.

As **Deactivation variable** a boolean project variable can be defined. At a rising edge on this variable the alarm creation for all alarms of the group will be deactivate, at a falling edge it will be re-activated.

Via button **Add** an alarm can be added to the group. A new line in the table window will be inserted and there the following parameters are to be set:

Expression: Enter here the project variable or an expression (e.g. "a + b") to which the alarm should refer. It is recommended to use the input assistant <F2> resp. the "Intellisense function" for an correct entry.

Configuration dialog Alarm group



Type: The alarm types listed in the following can be used. For each type regard the appropriate comment resp. the definitions to be done in the area beyond the table !

DIG=0 Digital alarm, active as soon as the expression gets FALSE.

DIG=1 Digital alarm, active as soon as the expression gets TRUE.

LOLO Analog alarm, active as soon as the value of the expression falls below the **Value** defined for **Alarm type LOLO**. You can define a tolerance (**Deadband**). As long as the expression value is within the dead band, no alarm will be activated, even if the LOLO-value has been falling below the limit.

LO corresponding to LOLO.

HI Analog alarm, active as soon as the expression exceeds the **Value** defined for Alarm type HIHI. You can define a tolerance (**Deadband**). As long as the expression value is within the dead band, no alarm will be activated, even if the HI value has exceeded the limit.

HIHI corresponding to HI.

DEV- Deviation from the target value; Alarm gets active as soon as the value of the expression falls below the value defined for **Alarm type DEV-** plus the percentage deviation. Percentage deviation = target value * (**deviation in %**) / 100.

DEV+ Deviation from the target value); Alarm gets active as soon as the value of the expression exceeds the value defined for **Alarm type DEV+** plus the percentage deviation. Percentage deviation = target value * (**deviation in %**) / 100.

ROC Rate of Change per time unit; Alarm gets active as soon as the expression deviates strongly from the previous value. The limit value for activating an alarm is defined by the number of value changes (**Rate of changes**) per second, minute or hour (**units per**).

Class: Choose the desired alarm class. The selection list will offer all classes which have been defined in the alarm class configuration before the last saving of the project.

Priority: Here you can define a priority level 0-152. 0 is the highest priority. The priority will impinge on the sorting of the alarms within the alarm table.

Message: Define here the text for the message box, which will appear in case of an alarm. This box must be confirmed by the user with OK, but this OK will not automatically acknowledge the alarm ! For confirming (acknowledge) the alarm you must access the alarm table. This is possible via the visualization element alarm table or via the date of the alarm entry in the table. This date can be read from a log file which can be created optionally.

Deactivation: Here a project variable can be entered, which at a rising edge will deactivate any creation of the alarm. Regard however, that this setting **will be overwritten** by the entry which might be found in the field 'Deactivation variable' ! (see above).

6.3.5 Alarm saving

For each alarm group a file can be defined, in which the alarm events are stored, if (!) a 'Save' action has been assigned to the class in the alarm class configuration dialog.

Select the alarm group in the configuration tree and open the dialog tab 'Alarm saving':

Configuration dialog 'Alarm saving'

The following definitions are possible:

Filepath: Directories path of the file which is defined in Filename; via button "..." you get the standard dialog for selecting a directory.

Filename: Name of the file which should save the alarm events (e.g. "alarmlog"). Automatically a file will be created which gets the name defined here plus an attached digit and which has the extension ".alm". The digit indicates the version of the log-file. The first file gets a "0"; each further file, which will be created according to the defined **File change event**, will be numbered with 1, 2 etc. (Examples: "alarmlog0.alm", "alarmlog1.alm").

File change event: Define here the event which will cause the creation of a new file for alarm-saving. Possible entries: **Never**, after one **Hour**, after one **Day**, after one **Week**, after one **Month**, at a rising edge of the variable defined in field **Triggervariable**, when the number of records in the file as defined in **Number of records** gets exceeded.

Triggervariable resp. **Number of records:** see above, File change event.

Delete old files after .. Hours: Number of days since the day of creation, after which all alarm log-files except from the actual one should be deleted.

The log-file (History) contains the following entries:

(See the column types and exemplary entries for two alarms)

Date/Time in DWORD	Date	Time	Event	Expression	Alarm type	Limit	Tolerance	current value	class	Priority	Message
1046963332	6.3.03	16:08:52	INTO	PLC_PRG.b	LO	-30	5	-31	Alarm_high	0	Temperature !
1046963333	6.3.03	16:08:53	ACK	PLC_PRG.n	HIHI	35			Warnng	9	Rising Temp. !

Example as it might look in the log-file:

1046963332,6.3.03 16:08:52,INTO,PLC_PRG.ivar5,HIHI,,,, 9.00,a_class2,0,

1046963333,6.3.03 16:08:53,INTO,PLC_PRG.ivar4,ROC,2,,, 6.00,a_class2,2,

1046963333,6.3.03 16:08:53,INTO,PLC_PRG.ivar3,DEV-,,,, -6.00,a_class2,5,

1046963334,6.3.03 16:08:54,INTO,PLC_PRG.ivar2,LOLO,-35,,3, -47.00,warning,10,warning: low temperature !

1046963334,6.3.03 16:08:54,INTO,PLC_PRG.ivar1,HI,20,,5, 47.00,a_class1,2,temperature to high !
Acknowledge !

6.3.6 'Extras' Menu: Settings

The dialog **Alarm configuration settings** opens on the command 'Extras' 'Settings' in the Alarm Configuration:

Category Date/Time:

Here you set the **date and time formatting** for the representation of the alarms in the log-file. Define the format according to the following syntax. Dashes and colons are to be set in inverted commas:

for date: dd-'MM'-'yyyy -> e.g. "12.Jan-1993"

for time: hh':'mm':'ss (12-hours format) -> e.g. "01:10:34" or HH':'mm':'ss (24-hours format) -> e.g. "13:10:34"

Language:

The usage of this dialog corresponds to that used for doing the language settings of a visualization object (see see User Manual for the CoDeSys Visualization: Language switching in visualization).

Choose here a language file (*.vis or *.tlt) which should be used for the alarm configuration texts when the language in CoDeSys is changed. Make sure that for this purpose the language file i.a. contains the translations for the text strings of the alarm configuration. See also: Chapter 4.3, Translate project into another language.

Alternatively you can define dynamic language switching by specifying a special language file in xml-format, like it is used for visualization elements. Note however that the actual settings only concern the display of the alarm messages in CoDeSys. If the alarm configuration is visualized by an alarm table element in a visualization object, there the separate visualization language settings will be valid!

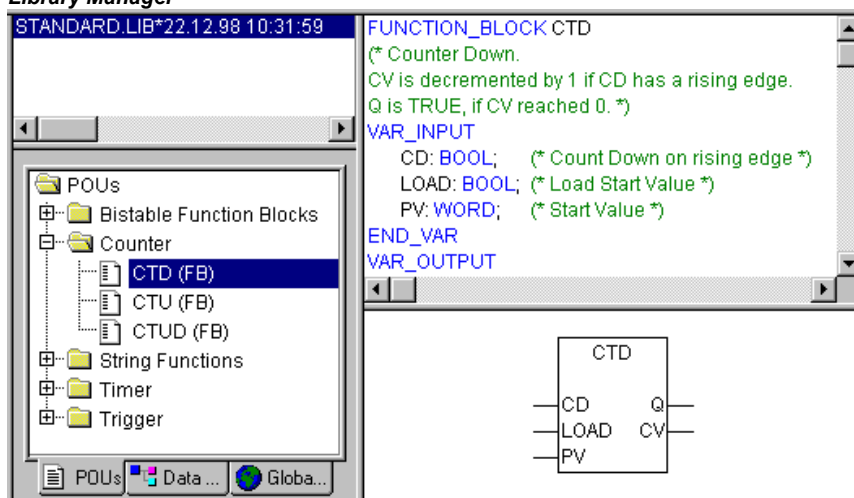
Online:

Deactivate alarm evaluation in online mode: If this online setting is set, there will be no alarm handling in online mode. This means that no alarm evaluation corresponding to the current alarm configuration will be done and no alarms will be displayed. This might be desired in certain environments in order to save computing time.

6.4 Library Manager

The library manager shows all libraries that are connected with the current project. The POU's, data types, and global variables of the libraries can be used the same way as user-defined POU's, data types, and global variables.

Library Manager



The library manager is opened with the **'Window' 'Library Manager'** command. Information concerning included libraries is stored with the project and can be viewed in the dialog 'Information

about external library'. To open this dialog select the corresponding library name in the library manager and execute the command 'Extras' 'Properties'.

Using the Library Manager

The window of the library manager is divided into three or four areas by screen dividers. The libraries attached to the project are listed in the upper left area.

In the area below that, depending on which register card has been selected, there is a listing of the **POUs** , **Data types**, **Visualizations** or **Global variables** of the library selected in the upper area.

Folders are opened and closed by double-clicking the line or pressing <Enter>. There is a plus sign in front of closed folders, and a minus sign in front of opened folders.

If a POU is selected by clicking the mouse or selecting with the arrow keys then the declaration of the POU will appear in the upper right area of the library manager; and in the lower right is the graphic display in the form of a black box with inputs and outputs.

With data types and global variables, the declaration is displayed in the right area of the library manager.

Standard Library

The library with "standard.lib" is always available. It contains all the functions and function blocks which are required from the IEC61131-3 as standard POUs for an IEC programming system. The difference between a standard function and an operator is that the operator is implicitly recognized by the programming system, while the standard POUs must be tied to the project (standard.lib).

The code for these POUs exists as a C-library and is a component of **CoDeSys**.

User-defined Libraries

If a project is to be compiled in its entity and without errors, then it can be saved in a library with the **'Save as'** command in the **'File'** menu. The project itself will remain unchanged. An additional file will be generated, which has the default extension ".lib". This library afterwards can be used and accessed like e.g. the standard library.

For the purpose to have available the POUs of a project in other projects, save the project as an **Internal Library *.lib**. This library afterwards can be inserted in other projects using the library manager.

If you have implemented POUs in other programming languages, e.g. C, and want to get them into a library, then save the project using data type **External Library *.lib**). You will get the library file but additionally a file with the extension "*.h". This file is structured like a C header file and contains the declarations of all POUs, data types and global variables, which are available with the library. If an external library is used in a project, then in simulation mode that implementation of the POUs will be executed, which was written with CoDeSys; but on the target the C-written implementation will be processed.

If you want to add licensing information to a library, then press button **Edit license info...** and insert the appropriate settings in the dialog 'Edit Licensing Information'. See the corresponding description at 'File' 'Save as...' resp. at License Management in CoDeSys.

'Insert' 'Additional Library'

With this command you can attach an additional library to your project.

The command opens the dialog for opening a file. If the currently set directory does not contain the desired library, you can select another directory in field **Library directory** where all directories will be offered, which are defined in 'Project' 'Options' 'Directories' 'Libraries' (File type "*.lib"). Choose the desired library - multiple selection is possible - and confirm with OK. The dialog will close and the library gets inserted to the Library Manager. Now you can use the objects of the library in the project like user-defined objects. [Library paths](#)

Regard which libraries directories are currently defined in the project options (see Chapter 4.2, category 'Directories'). If you insert a library from a directory which is not defined there, the library will be entered with the respective path.

Example: You insert library standard.lib from directory "D:\codesys\libraries\standard".

If this directory is defined in the project options, the entry in the library manager will be: "standard.lib <date and time of file>".

If in the project options there is just defined a directory "D:\codesys\libraries", then the entry in the library manager will be: "standard\standard.lib <date and time of file>".

If no matching directory at all is defined in the project options, then the complete path will be entered: "D:\codesys\libraries\standard\standard.lib <date and time of file>".

When re-opening the project the libraries will be searched according to entries in the library manager. So for example, if just the library file name is entered there, the library will be searched in the libraries directories defined in the project options.

If libraries are not found when opening a project, you will be asked whether you want to change the library directory defined in the project options. If you say No, a dialog will open providing information on the libraries not found and the respective entries in the Library Manager will be displayed red-colored. In this case you can select a red entry and choose command **Search** from the context menu. Thus you will get the dialog for opening a file where you can browse for the missing library and reload it immediately.

Licensing

As soon as you include a library for which a license is needed and no valid license is found, you may get a message that the library is only available in demo mode or that the library is not licensed for the currently set target. You can ignore this message at that time or start appropriate actions concerning the license. An invalid license will produce an error during compile ('Project' 'Build'). In this case a double-click on the error message resp. <F4> will open the dialog 'License information' where you can start the appropriate actions guided by a wizard.

Remove Library

With the 'Edit' 'Delete' command you can remove a library from a project and from the library manager.

'Extras' 'Properties'

This command will open the dialog 'Information about internal (resp. external) library'. For internal libraries you will find there all data, which have been inserted in the Project Info (where applicable including the license information) when the library had been created in CoDeSys. For external libraries the library name and library path will be displayed.

6.5 Log

The log stores in chronological order actions that occur during an Online session. For this purpose a binary log file (*.log) is set up. Afterward, the user can store excerpts from the appropriate project log in an external log.

The log window can be opened in either Offline or Online mode and can thus serve as a direct monitor online.

'Window' 'Log'

To open, select the menu item 'Window' 'Log' or select entry 'Log' in the Resources tab.

In the log window, the filename of the currently displayed log appears after **Log:**. If this is the log of the current project, the word "(Internal)" will be displayed.

Registered entries are displayed in the log window. The newest entry always appears at the bottom.

Only actions belonging to categories that have been activated in the 'Filter' field of the menu 'Project' 'Options' 'Log' will be displayed.

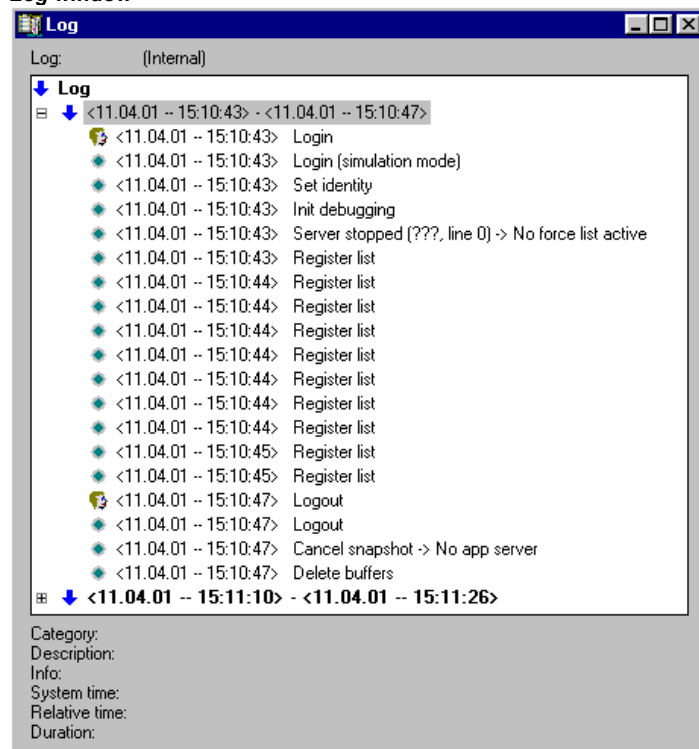
Available information concerning the currently selected entry is displayed below the log window:

Category: The category to which the particular log entry belongs. The following four categories are possible:

- User action: The user has carried out an Online action (typically from the Online menu).
- Internal action: An internal action has been executed in the Online layer (e.g. Delete Buffers or Init debugging).
- Status change: The status of the runtime system has changed (e.g. from Running to Break, if a breakpoint is reached).

Exception: An exception has occurred, e.g. a communication error.

Log window



Description: The type of action. User actions have the same names as their corresponding menu commands; all other actions are in English and have the same name as the corresponding OnlineXXX() function.

Info: This field contains a description of an error that may have occurred during an action. The field is empty if no error has occurred.

System time: The system time at which the action began, to the nearest second.

Relative time: The time measured from the beginning of the Online session, to the nearest millisecond.

Duration: Duration of the action in milliseconds.

Menu Log

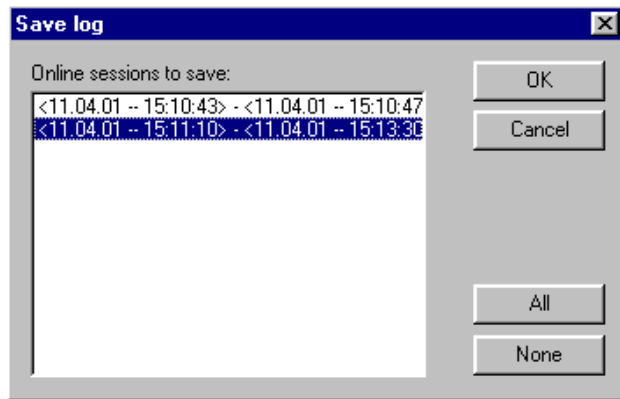
When the log window has the input focus, the menu option **Log** appears in the menu bar instead of the items 'Extras' and 'Options'.

The menu includes the following items:

Load... An external log file *.log can be loaded and displayed using the standard file open dialog.

The log that is present in the project will not be overwritten by the command. If the log window is closed and later opened again, or a new Online session is started then the version that is loaded will again be replaced by the project log.

Save... This menu item can only be selected if the project log is currently displayed. It allows an excerpt of the project log to be stored in an external file. For that, the following dialog will be displayed, in which the Online sessions to be stored can be selected:



After successful selection, the standard dialog for storing a file opens ('Save Log').

Display Project Log This command can only be selected if an external log is currently displayed. It switches the display back to the project log.


Storing the project log

Regardless of whether or not the log is stored in an external file (see above), the project log is automatically stored in a binary file entitled <projectname>.log. If a different path is not explicitly given in the 'Project' 'Options' 'Log' dialog, the file is stored in the same directory as that in which the project is stored.

The maximum number of Online sessions to be stored can be entered in the 'Project' 'Options' 'Log' dialog. If this number is exceeded during recording, the oldest session is deleted to make room for the newest one.

6.6 PLC Configuration

6.6.1 Overview

The  PLC Configuration is found as an object in the register card **Resources** in the Object Organizer. With the PLC Configuration editor, you must describe the hardware the opened project is established for. For the program implementation, the number and position of the inputs and outputs is especially important. With this description, **CoDeSys** verifies whether the IEC addresses used in the program also actually exist in the hardware.

The base of working in the configuration editor is/are the configuration files (*.cfg; see below **Note concerning version compatibility**) and the device files (.e.g. *.gsd, *.eds). These are stored in the directory which is defined in the target file (see Target Settings) and are read when the project is opened in **CoDeSys**. **You can add files to this directories at any time.**

The configuration file *.cfg describes a basic configuration, which is mapped in the configuration editor, and it defines to which extent the user can customize this configuration in the editor.

Attention: As soon as the underlying configuration file (*.cfg) has been modified, you have to redo the configuration in CoDeSys!

Note concerning version compatibility: In CoDeSys V2.2 a new format was implemented for the PLC Configuration. From that version on the basic configuration files have to use the extension *.cfg. In contrast the configuration editor in former CoDeSys versions needed configuration files with an extension *.con. But: The target system might define that the "old" configurator should be used further on, even when an old project is opened in V2.2 or higher (see remarks at the end of this chapter). This avoids the need of creating new configuration files, the *.con-files can be used further on. If the target system does not support the old configurator, you can convert the old PLC

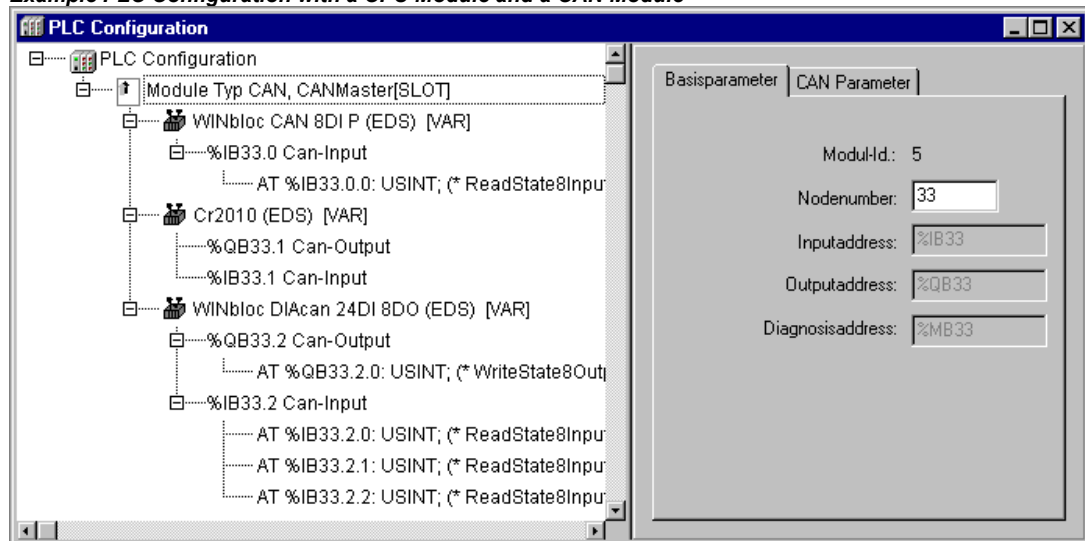
Configuration, which is stored in the project, to the new format, if (!) an appropriate new *.cfg-file has been provided. See more details in Chapter 6.6.3 'Extras' 'Convert'.

The **CoDeSys** configuration editor allows configuring I/O modules as well as CAN and Profibus modules.

If supported by the target system, there is the possibility to get information from the PLC: 1. a scan of the actual hardware structure which can directly be used in the PLC Configuration, 2. diagnosis messages which will be displayed as messages in CoDeSys, 3. status information which will be displayed in the PLC Configuration dialog

After the final customization by the user a binary image of the configuration is sent to the PLC:

Example PLC Configuration with a CPU Module and a CAN Module



The PLC Configuration is displayed in the editor in tree structure and can be edited using menu commands and dialogs. The configuration contains input and/or output elements and also management elements which themselves also have sub elements (for example, a CAN-bus or a digital input card with 8 inputs).

For inputs and outputs, symbolic names can be assigned. The IEC-address where this input or output can be accessed is then located behind the symbolic name.

Possibly the currently used target system defines that the **CoDeSys V2.1 PLC Configuration** is to be used. In this case for help on working with the PLC Configuration see the user documentation on CoDeSys V2.1, but regard the following extensions in the configurator:

CAN configurator:

Option "Create all SDO's" in dialog 'CAN Parameters' of a CAN module (see Chapter 6.6.8). Attention: The SDO's now are always created according to the mechanism used in the V2.3 configurator; thus the results may differ from that you got previously.

Edit field "Device-Type" in dialog 'CAN settings' of a CanDevice (see Chapter 6.6.9).

Profibus configurator:

The selection list of the Profibus modules (see Chapter 6.6.7) available for getting inserted now is sorted alphabetically by module names.

6.6.2 Working in the PLC Configuration

The configuration editor is divided up in two parts. In the left window the **configuration tree** is displayed. Structure and components of the tree result primarily (Standardconfiguration) from the definitions found in the configuration file, but can be modified by the additional adaptation which is

done by the user in the CoDeSys PLC Configurator. In the right window the currently available **configuration dialogs** are shown on one or several tabs.

The right part of the window is per default visible, but can get faded out via the menu item **'Extras' 'Properties'**.

On top of the configuration tree there is the entry of the **"root" module** with a name, which has been defined in the configuration file.

Below you find hierarchically indented the other elements of the configuration: Modules of different types (CAN, Profibus, I/O), channels or bit channels.

The configuration editor is divided up in two parts. In the left window the **configuration tree** is displayed. Structure and components of the tree result primarily (Standardconfiguration) from the definitions found in the configuration file, but can be modified by the additional adaptation which is done by the user in the CoDeSys PLC Configurator. In the right window the currently available **configuration dialogs** are shown on one or several tabs.

The right part of the window is per default visible, but can get faded out via the menu item **'Extras' 'Properties'**.

On top of the configuration tree there is the entry of the **"root" module** with a name, which has been defined in the configuration file.

Below you find hierarchically indented the other elements of the configuration: Modules of different types (CAN, Profibus, I/O), channels or bit channels.

Selecting of elements

For selecting elements, click the mouse on the corresponding element, or, using the arrow keys, move the dotted rectangle onto the desired element.

Elements that begin with a plus sign are organization elements and contain sub elements. To open an element, select the element and double-click the plus sign or press <Enter>. You can close opened elements (minus sign in front of the element) the same way.

Insert elements, 'Insert' 'Insert element', 'Insert' 'Append subelement'

Depending on the definitions in the configuration file(s) and on the available device files, which have been read when the project was opened, a basic composition of elements is automatically positioned in the configuration tree. If one of those elements is selected, further elements may be added if this is allowed by the definitions in the configuration file and if the needed device files are available:

'Insert' 'Insert element': An element can be selected and inserted before the element which is currently marked in the configuration tree.

'Insert' 'Append subelement': An element can be selected and inserted as subelement of the element which is currently marked in the configuration tree. It will be inserted at the last position.

The most important commands are found in the context menu (right mouse button or <Ctrl>+<F10>).

Please note: If supported by the target system, a scan of the existing hardware can be used for the inserting of modules in the CoDeSys PLC Configuration.

Replacing/switching Elements, 'Extras' 'Replace element'

Depending on the definition in the configuration file, the currently selected element may be get replaced by an other one. The same way it may be possible to switch channels, which are set up in a way that they can be used as input or as output elements. Use the command 'Extras' 'Replace element'

Symbolic names

Symbolic names for modules and channels can be defined in the configuration file. In this case they will be shown in the configuration editor before the 'AT' of the IEC address of the respective element. In the configuration file also is defined whether the symbolic name can be edited or inserted in the configuration editor. To enter a symbolic name, select the desired module or channel in the

configuration tree and open a text field by a mouse-click on the 'AT' before the IEC address. In the same manner you can edit an existing symbolic name after a double-click on the name. Please regard that allocating a symbolic name corresponds with a valid variable declaration !

Export/Import of modules

If a module is defined as being "exportable" in the configuration file (*.cfg), in the context menu you will find the commands 'Export module' and 'Import module' when the module is selected in the configuration tree.

With command 'Export module' the dialog for selecting a file will be opened. Here you can define a file to which the module will be exported in XML format together with all submodules and their configuration. This file can be imported in another PLC configuration via command 'Import module', if there an appropriately defined module is selected in the configuration tree.

Thus in an easy way the configuration tree of a particular module can be transferred to another PLC configuration.

6.6.3 General Settings in the PLC Configuration

Select the entry 'PLC configuration' ('root' module) at top of the configuration tree. Thereupon the dialog 'Settings' is shown in the right part of the window. The following options can be activated:

Calculate addresses: Each newly inserted module automatically is allocated with an address, which results from the address of the module inserted before plus the size of this address. If a module is removed from the configuration, the addresses of the following modules are adjusted automatically. When the command 'Extras' 'Compute addresses' is executed, all addresses starting at the selected node (module) will be recalculated.

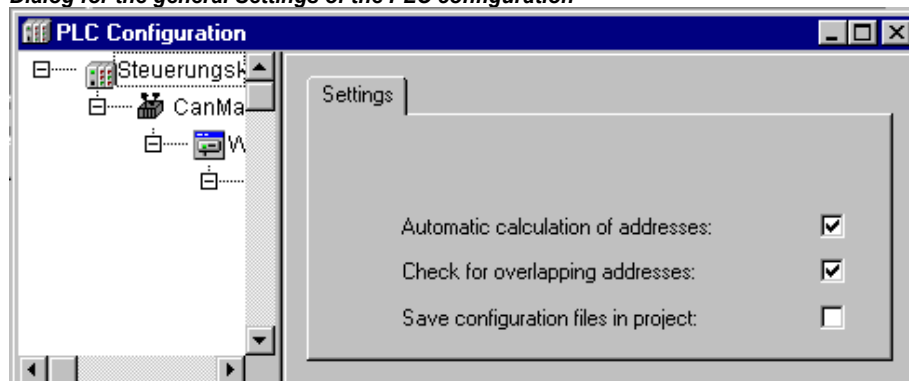
Check for overlapping addresses: At compilation the project will be checked for overlapping addresses and a corresponding message will be displayed.

Save configuration files in project: The information which is contained in the configuration file(s) *.cfg and the device description files, which underlie the current PLC Configuration, will be saved in the project.

Thus (if it is not defined by the configuration file, that **always the standard configuration** should be reloaded !), the configuration as set up by the user will be kept in the project, even if configuration files are not found when the project is re-opened. Keep in mind that in such a case the complete project specific configuration **will get lost**, if the here described option is not activated !

By saving the configuration information with the project these also will be kept at a target change. But regard in this case, that the new target might bring own configuration files which then will be regarded additionally.

Dialog for the general Settings of the PLC configuration



The global mode of addressing (flat addresses / addresses depending on Id) in the PLC configuration is defined in the configuration file.

Recalculation of Module addresses, 'Extras' 'Compute addresses'

If the option "Calculate addresses" is activated in the dialog 'Settings' of the PLC configuration editor , then the command 'Extras' 'Compute addresses' will start to recalculate the addresses of the modules. All modules starting with the one, which is currently selected in the configuration tree, will be regarded.

Add configuration file

Use this command in the 'Extras' menu to add a further file to the configuration files of the project. These are the files which are found in the directory path(es) specified in the project options, category 'Directories', input field 'Configuration files'.

The dialog **Select configuration file** will be opened, where you can set a filter for CAN- (*.eds,*. dcf), Profibus- (gsd*.*) , configuration- (*.cfg)-files or all files (*.*). After having selected the desired file a check will be done, whether the file is already found in one of the defined directories for configuration files. In this case an appropriate message will appear and the file cannot be added. If a cfg-file is selected, in each case an dialog will open where you get information on what to do.

If the file can be added, the dialog **Select configuration directory**, where all configuration directories currently defined for the project will appear in a selection list. Choose the directory where the file whereto the file should be copied. After having confirmed this selection by pressing button OK, the dialog will close and the file immediately will be available in the PLC configuration.

Return to standard configuration, 'Extras' 'Standard configuration'

The command 'Extras' 'Standardconfiguration' can be used to restore the original PLC configuration, which is defined by the configuration file *.cfg and saved in the project.

Attention: In the configuration file *.cfg it might be defined that the standard configuration should be restored at **each** reopening of the project. In this case all adaptations of the configuration done by the user will get lost !

Converting of old PLC configurations, 'Extras' 'Convert'

This command is available in the menu 'Extras' if you open a project containing a PLC configuration, which was created with an older **CoDeSys** version than V2.2. If all needed configuration files are available, the command 'Convert' will transfer the existing configuration into the format of the actual PLC configuration. A dialog will open which asks "Convert the configuration to the new format ? Attention: Undo is not possible !" You can select **Yes** or **No**. If you close the dialog with **Yes**, the configuration editor will be closed also. Reopen it and you will see the configuration in the new format. Be aware that after having converted the old format cannot get restored anymore !

6.6.4 Custom specific parameter dialog

The parametrizing possibilities of the configurator can be expanded by the use of an application-specific DLL which is an individual dialog. This 'Hook'-DLL must be in that directory which contains the configuration file and then can be linked by an entry in the configuration file.to a module or channel. If done so, for the concerned modules the standard dialog 'Module parameters' will be replaced by a dialog defined in the DLL.

Example of an application-specific parameter dialog (Custom Parameters)

Custom Dialog

PgmVersion: 0

LibDir: d:\codesys\lib

ObjectId: 1

SectionName: Module.CPU1

PrmCount: 9

Definierte Parameter

1, 10000, Parameter.RateType, XRate, 10
2, 10000, Parameter.RateType, Yrate, 10
3, 10001, Parameter.BoolType, EnableDiags, Yes
4, 10007, Parameter.NameType, Namen, HugoType
5, 107659, Parameter.FileType, DumpFile, D:\CoDeSys\Projekte\Dump.bin
6, 1343, Parameter.NameString, IDString, abc def
7, 1347, Parameter.RealType, Realvalue, 1.876
8, 1348, Parameter.Bool, Boolvalue, TRUE
9, 138740, Parameter.IntTypeHex, Test, 16#A

Param1 (Value): 10

6.6.5 Configuration of an I/O Module

Base parameters of an I/O Module

If an I/O module is selected in the configuration tree, the dialog 'Base parameters' is displayed with the following entries:

Module id: The Module id is a unique identifier of the module within the entire configuration. It is defined by the configuration file and it is not editable in the configuration editor.

Node id: The Node ID can be defined by an entry in the configuration file or – if there is no entry – by the position of the module in the configuration structure.

Input address, Output address, Diagnostic address: Addresses for Input- and Output respectively for the storage of diagnosis data.

These addresses refer to the module. It depends on the general settings, which addresses are already predefined, which address mode is valid and whether the addresses can be still edited here.

Comment: Here additional information text on the module can be entered.

Load module description: If this option is deactivated, the module will not be regarded at a download of the project. Per default the option is activated and it is defined in the configuration file *.cfg whether it is visible and editable in the configuration dialog.

Do not adapt address automatically: This option is only available if defined by the configuration file. If it is activated, the module will not be regarded in case of a recalculation of the addresses. (Default: Option is deactivated.)

Base parameters Dialog for an I/O Module

The screenshot shows a dialog box titled "Base parameters Dialog for an I/O Module". It contains the following fields and options:

- Module id: 1
- Node id:
- Input address:
- Output address: Do not adapt address automatically
- Diagnostic address:
- Comment:
- Load module description:

The diagnosis in the PLC configuration:

The field **Diagnostic address** serves for specifying a marker address to which the diagnosis information (bus state) will be written automatically.

For normal I/O modules it depends on the special hardware configuration how the diagnosis will be handled. For bus systems like CAN or Profibus DP the diagnosis works like described in the following: From the given diagnosis address onwards there will be stored various information concerning the **structure GetBusState** which is part of a corresponding library delivered by the manufacturer (e.g. BusDiag.lib by 3S - Smart Software Solutions). All bus modules get a request to fill the diagnosis structure in a cyclic sequence each time when an IEC task has written or read process data to/from the modules.

- **Diagnosis analysis via function block *DiagGetState*:**

As soon as at least one module in the bus system produces an error, the specific diagnosis information can be read using the function block *DiagGetState* which is also part of the above mentioned library. This function is only available for bus masters, which have been configured within the CoDeSys PLC configuration!

In order to be able to analyze the diagnosis data given in structure GetBusState, a variable of type GetBusState must be assigned to the marker address (diagnostic address), e.g.:

```
BusState AT %MB12 : GetBusState;
```

See in the following the input and output parameters of the function block *DiagGetState*. Define an instance of this function block in your CoDeSys project to read the diagnosis information for a specific bus module:

Input variables of *DiagGetState*:

ENABLE: BOOL;	At a rising edge of ENABLE the function block starts working
DRIVERNAME:POINTER TO STRING;	Name of the driver (address of the name) to which the diagnosis request should be sent. If here is entered a 0, the diagnosis request will be forwarded to all present drivers.
DEVICENUMBER:INT;	Identification of the bus which is managed by the driver. E.g.: the Hilscher driver can handle up to 5 cards (busses). The index is 0-based.
BUSMEMBERID:DWORD ;	Unique bus-/driver specific identification of the busmodule (E.g.: for a CANopen-card this is the NodeID, for a PB-DP card this is the station address of the participant etc.)

Output variables of DiagGetState

READY:BOOL ;	TRUE: the work on the diagnosis request has been terminated
STATE:INT;	If READY = TRUE then STATE gets one of the following values which define the actual state of the function block: -1: invalid input parameter (NDSTATE_INVALID_INPUTPARAM:INT;) 0: function block does not work (NDSTATE_NOTENABLED:INT;) 1: function block is just reading the diagnosis info (NDSTATE_GETDIAG_INFO:INT;) 2: diagnosis info is now available (NDSTATE_DIAGINFO_AVAILABLE:INT;) 3: no diagnosis info is available (NDSTATE_DIAGINFO_NOTAVAILABLE:INT;)
EXTENDEDINFO: ARRAY[0..129] OF BYTE;	Up to 100 Bytes manufacturer specific diagnosis data of the bus.

- **Alternative diagnosis analysis via function block *DiagGetBusState*:**

If a diagnostic address is specified in the configuration, the bus state will be updated in the background. If no diagnostic address is specified or if no configuration has been done in CoDeSys, then – if supported by the target system – the function block *DiagGetBusState* can be instantiated and explicitly called in order to get diagnosis data from the bus. This function block like *DiagGetState* also must be available in a manufacturer specific library, e.g. *BusDiag.lib* by 3S - Smart Software Solutions GmbH.

The module – except for *BusMemberID* – uses the same input and output parameters. The parameters STATE and EXTENDEDINFO however have the following, different meaning:

STATE:INT;	If READY = TRUE then STATE gets one of the following values which define the actual bus state: BUSSTATE_BUSOK (Bus is running correctly) BUSSTATE_BUSFAULT (Bus error) BUSSTATE_BUSNOTCOMMUNICATING (Communication error) BUSSTATE_BUSSTOPPED (Bus is stopped)
EXTENDEDINFO:ARRAY[0..129] OF BYTE;	Manufacturer specific diagnosis data of the bus member; only the first three bits of the byte are used: Bit 0: bus member is configured Bit 1: bus member is active Bit 2: bus member reports an error, detailed information can be get via <i>DiagGetState()</i> *)

Module parameters / Custom parameters of an I/O Module

Module parameters Dialog

Index	Name	Value	Default	Min.	Max.
1	XRate	10	10		
2	Yrate	10	10		
3	Enable...	Yes	Yes		
4	Namen	HugoType	HugoType		
5	DumpFi...	D:\CoDeSys\Projekte\ID...	D:\CoDeSys\Proj...		
6	IDString	abc def	abc def		
7	Realval...	1.876	1.876	-1.1	2.9876
8	Boolval...	TRUE	TRUE		
9	Test	16#A	16#A	16#A	16#FF

In this dialog the parameters which are given by the device file are shown. Only the column 'value' is editable.

Index: The Index is a consecutive digit (i), which numbers through all the way the parameters of the module.

Name: Name of the parameter

Value : Value of the parameter, editable

Initially the default is displayed. Values can be set directly or by means of symbolic names. If the entries in the configuration file are not set to 'Read Only', they can be edited. To do that click on the edit field respectively select on of the entries in a scroll list. If the value is a file name, you can open the dialog 'Open file' by a double-click and browse for another file there.

Default: Default value of the parameters

Min.: minimum value of the parameter (only if no symbolic names are used)

Max.: maximum value of the parameter (only if no symbolic names are used)

A tooltip may give additional information on the currently marked parameter.

Instead of the Module parameters dialog there might be a customer specific dialog. This is due to the fact, that such a dialog is linked by an entry (Hook-DLL) at the module definition in the configuration file.

6.6.6 Configuration of a Channel

Base parameters of a channel

Channel-Id: Globally unique identifier of the channel

Class: Defines whether the channel is used as input (I), output (Q), or as input and output (I&Q), or whether it is switchable (I|Q). If the channel is switchable, this can be done by the command 'Extras' 'Replace element'.

Size: Size of the channel [Byte]

Default identifier: Symbolic name of the channel

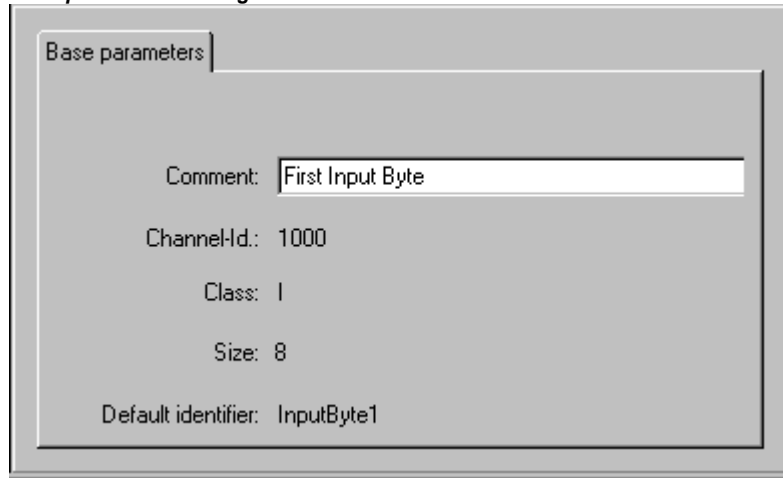
The name of the channel is defined in the configuration file. Only if it is allowed by the definition of the father module, the name of the channel can be edited in the configuration tree.

Comment: Additional information on the channel

In the edit field a comment can be inserted or modified.

Address: This edit field only will be available if it was activated by an entry in the configuration file. Insert the desired address for the channel.

Base parameters dialog for a channel



Channel parameters

Corresponding to the Module parameters dialog the Channel parameter dialog is used to display and modify the parameters of a channel: **Index**, **Name**, **Value**, **Default**, **Min.**, **Max**. This dialog also can be replaced by a customer specific dialog 'Custom Parameters'.

Bitchannels

Bitchannels are automatically inserted, when a channel is defined with an entry `CreateBitChannels=TRUE` in the configuration file.

The Base parameters dialog of bitchannels just contains the field **Comment**.

6.6.7 Configuration of Profibus Modules

CoDeSys supports a hardware configuration corresponding to the profibus DP standard. In the profibus system you find master and slave modules. Each slave is provided with a parameter set by ist master and supplies data on request of the master.

A PROFIBUS DP system consists of one or several masters and their slaves. First the modules must be configured so that a data exchange over the bus is possible. At the initialization of the bus system each master parameterizes the slaves which are assigned to it by the configuration. In a running bus system the master sends and/or requests data to/from the slaves.

The configuration of the master and slave modules in **CoDeSys** is based on the gsd files attached to them by the hardware manufacturer. For this purpose all gsd-files which are stored in the configuration files directory (see Chapter 4.2, Project Options: Directories) will be considered. The modules described by a gsd file can be inserted in the configuration tree and their parameters can be edited there.

Below a master there can be inserted on or several slaves.

If a DP master is selected in the configuration tree, the following dialogs will be available in the right part of the configuration: Base parameters, DP Parameter, Bus parameters, Module parameters.

If a DP slave is selected, which is inserted below a DP master, the following dialogs can be available (depending on the definitions in the configuration file): Base parameters, DP Parameter, Input/Output, User parameters, Groups, Module parameters. Depending on the settings in the configuration file the dialog "DP Parameter" might have another title .

If a DP slave is inserted on the level of a master, the following dialogs are available for configuration: Base parameters, DP parameters, Input/Output, Module parameters.

Base parameters of the DP master

The Base parameters dialog of a DP master matches that of the other modules (see chapter 6.6.5, 'Base parameters of an I/O Module').

Module parameters of the DP master

The Module parameters dialog of a DP master matches that of the other modules: The parameters assigned to the master in addition to the DP and bus parameters in the configuration file are displayed here and the values can be edited in the standard case (see Chapter 6.6.5 'Module parameters of an I/O Module').

DP parameters of the DP master

This dialog shows the following parameters extracted from the device file of the DP master (The dialog might have a different title, which is defined in the configuration file):

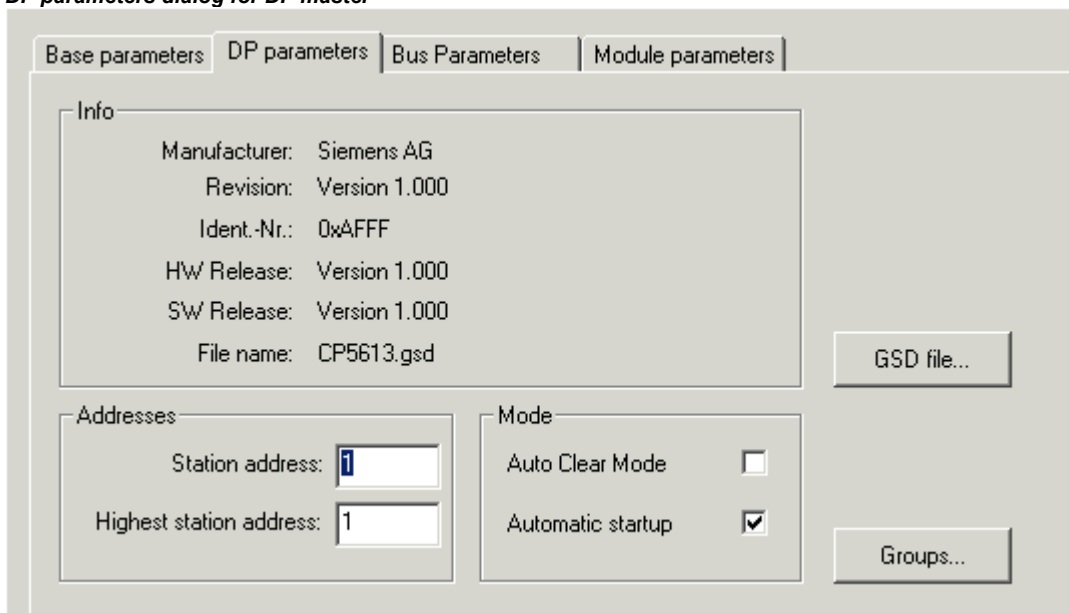
- Info** **Manufacturer, GSD Revision, ID** (identification number), **HW Release** and **SW Release** (hard- and software version), **GSD-Filename**

- Addresses** **Station address:** The allowable range extends from 0 to 126. Each device newly inserted on a bus line is automatically provided the next higher address. (note: Address 126 is the default DP slave address). Manual entry is possible; addresses are tested for duplication.
Highest station address: The highest station address (**HSA**) assigned on the bus is displayed. Here, a lower address can also be entered in order to narrow the GAP range (that is, the address range within which the search for newly-active devices is carried out).

- Mode** **Auto Clear Mode:** If this option is activated, the outputs of the slaves will be switched to a save state in error case and the master will change from state "operate" to "clear".
Automatic Startup: If this option is not activated, the master must be started manually. However it depends on the driver, whether the setting is evaluated.

The GSD file pertaining to a device can be opened and examined using the **GSD File** button.

DP parameters dialog for DP master



The **Groups** button leads to the 'Group properties' dialog. The Group properties pertain to the slaves assigned to the master. Up to eight groups can be set up. For each group, enter whether it is to operate in **Freeze mode** and/or **Sync mode**. By assigning slaves (see 'Properties of the DP slave' 'Group assignment') to various groups, data exchange from the master can be synchronized via a global control command. With a Freeze command, a master instructs a slave or a group to „freeze“ inputs in their instantaneous state and to transfer these data in the following data exchange. With a Sync command, the slaves are instructed to synchronously switch to output at the next Synch command all data received from the master following the first command.

To switch the Freeze and Sync options for a group on/off, please click with the left mouse button on the appropriate location in the table to place or remove an „X“ next to the desired option, or use the right mouse button to activate or deactivate the option via a context menu. In addition, you can edit the group name here.

DP parameters of the DP master / Group Properties dialog

Group Name	Sync. Mode	Freeze Mode
Gr 1	X	X
Gr 2		X
Gr 3	X	X
Gr 4		X
Gr 5	X	X
Gr 6	X	X
Gr 7		X
Gr 8	X	

Bus parameters of the DP master

The bus parameters describe the timing of the communication. If the option **Use defaults** is activated, then the parameter values will be calculated automatically depending on the **Baudrate** set by the user and the settings given by the GSD files.

Attention: : The automatically calculated values are just approximated values !

Bus parameters of the DP master

Parameters	Value	Unit
Slot Time (TSL)	0	tBit
Min.Station Delay (min TSDR)	0	tBit
Max.Station Delay (max TSDR)	0	tBit
Quiet Time (TQUI)	0	tBit
Setup Time (TSET)	0	tBit
Target Rotation Time (TTR)	0	tBit
Gap Update Factor	0	
Max. Retry Limit	0	
Min. Slave Interval	0	100 µs
Poll Timeout	0	10 ms
Data Control Time	0	ms

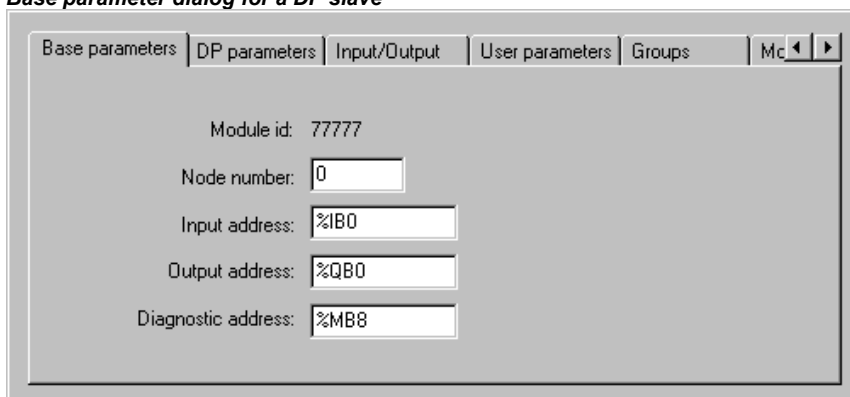
All parameters can also be edited manually.

Baud rate	The entries already present in the GSD file are available for selection, but only a transmission rate supported by all slaves can be entered.
Use defaults	If the option is activated, the entries made in the 'Bus parameters' dialog will be optimized with respect to the specifications in the GSD files; it is only possible to edit the values if the option is deactivated. Important: The values calculated automatically are only rough approximate values.
Slot Time	Maximum time during which the master waits, after sending a request message, for the receipt of the first character of the slave's reply message
Min.Station Delay	min. TSDR (in tbit): minimum reaction time after which a station on the bus may reply (min. 11 tBit)
Max.Station Delay	max. TSDR (in tbit): maximum time span within which a slave must reply.
Quiet Time	TQUI (in tbit): idle period which must be taken into account during conversion of NRZ (Non Return to Zero) signals to other codings (switchover time for repeater)
Target Rotation Time	TTR (in tbit): token cycle time setting; projected time interval in which a master should receive the token. Result of the sum of the token stop times of all masters on the bus.
Gap Update Factor	GAP update factor G: number of bus cycles after which the master's GAP (address range from its own bus address to the address of the next active station) is searched for an additional, newly inserted active station.
Max. Retry Limit	maximum number of repeated request attempts by the master when it has not received a valid response from the slave
Min. Slave Interval	Time between two bus cycles in which the slave can process a request from the master (time basis 100µs). The value entered here must be checked against the respective specifications in the slave's GSD file.
Poll Timeout	Maximum time after which the master's reply by a master-master communication must be retrieved by the requester (Class 2 DP master) (time basis 1 ms).
Data Control Time	Time in which the master reports its status to the slaves assigned to it. At the same time, the master monitors whether at least one data exchange each has taken place with the slaves within this period, and updates the Data_Transfer_List.
Watchdog Time	Time value for the access monitoring (watchdog). Setting is currently not supported (fixed-set to 400 ms)

Base parameters of a DP slave

The Base parameters dialog of a DP slave matches that of the other modules (see chapter 6.6.5, 'Base parameters of an I/O Module').

Base parameter dialog for a DP slave



DP parameters of a DP slave

This dialog shows the following parameters extracted from the device file of the DP slave (The dialog might have a different title, which is defined in the configuration file):

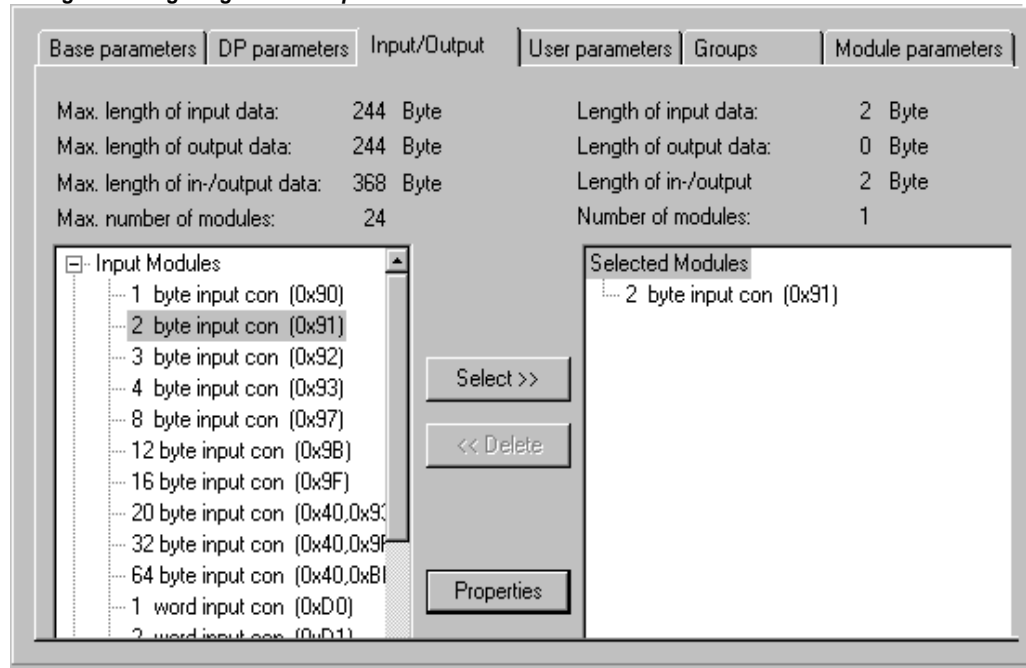
DP parameters dialog for a DP slave

Info	Manufacturer , GSD Revision , HW Release and SW Release (hard- and software version), GSD-Filename , Slave type
Standard parameter	<p>Identnumber: Unique identnumber assigned by the PNO for this device type. Allows unambiguous reference between a DP slave and the corresponding GSD file.</p> <p>TSDR (Tbit*): Time Station Delay Responder: Reaction time, the earliest time after which the slave is permitted to respond to the master. (min. 11 TBit) * TBit: Time unit for transfer of a bit on the PROFIBUS; Reciprocal value of the transmission rate; e.g. 1 TBit at 12MBaud=1/12.000.000 Bit/sek=83ns</p> <p>Lock/Unlock: Slave is locked or released to other masters: 0: min.TSDR and slave-specific parameters may be overwritten 1: Slave released to other masters, 2: Slave locked to other masters, all parameters are accepted; 3: Slave released to other masters</p>
Identification	Station address (see 'Properties of the DP masters'), Station name (matches device name, editable)
Activation	Slave is active/inactive in current configuration. If activation is not selected, configuration data will still be transferred to the coupler on Download, but not data exchange occurs over the bus.
Watchdog	If Watchdog Control is set active, the entered Watchdog time applies (access monitoring, basis 10 ms). If the slave has not been accessed by the master within this time, it is reset to its initialization state.

You can inspect the corresponding GSD file via the **GSD-File** button.

Input/Output of a DP slave

Dialog for configuring the in-/outputs of a DP slave



The way in which the configuration of a DP slave is done, depends on whether it is a 'modular' or a 'non-modular', 'fix' slave.

The selection of the module for a modular slave is done like described in the following:

In the list on the left part of the dialog select the desired input- or output-module and press button **Select** to get it into the window on the right hand. Wrong entries in that window can be corrected via the button **Delete**. Inserted modules will be immediately displayed in the configuration tree. If they are selected there, the appropriate dialog **Profibus Modul** will be available, showing the input-, output- and diagnosis address of the module. If you select a channel which has been inserted with the module, the dialog **Profibus Channel** will open, showing the address of the channel. For each of these both dialogs there might be defined a different title by a setting in the configuration file.

As the maximum data lengths specified in the GSD-file (**Max. length of input data**, **Max. length of output data**, **Max. length of in-/output data**) and the maximum number of modules (**Max. number of modules**) must be respected, this information is displayed in both module lists. The left block displays the maximum possible values for the device, the right the values resulting from summing the values selected in the configuration. If the maximum values are exceeded, an error message is issued.

The dialog lists in the left window all the in- and output modules available in the slave's GSD-file, while the right window contains the configuration currently selected for this device as it relates to in- and outputs.

If this is a modular slave (a device that can be equipped with various I/O modules), the selection proceeds as follows: In the left-hand list, the desired in- or output module is selected by mouse-click and copied into the right window using the **Select >>** button. Incorrect entries can be corrected by selecting the undesired module in the right window and pressing the **Delete** button.

This kind of selection is not possible for non-modular slaves. These directly enforce a closed display of their in- and outputs in the right window. Undesired modules can then be removed by selecting and using **Delete**.

The **Properties** button leads to the 'Module properties' dialog for the in- or output module currently selected in the left or the right list. It shows the **Name**, the **Config** (module description coding according to PROFIBUS standard) and the **in-** and **output lengths** of the module in **bytes**. If the module description in the GSD file contains specific parameters in addition to the standard set, these

are listed here with their values and range of values. If the **Symbolic names** option is activated, the symbolic names are then used.

User parameters of a DP slave

User parameter dialog for a DP slave

The dialog shows the 'User parameters' tab. At the top, it indicates 'Length of user parameters in bytes: 15' and a checked 'Symbolic names' checkbox. Below is a table with three columns: Parameters, Value, and Allow.

Parameters	Value	Allow
"Register-Interface"	wird nicht genutzt	Bit(0)
"RESET bei Klemmenbus-Fehler"	POWER ON RESET	Bit(0)
"Klemmenbus-Diagnose"	ausgeschaltet	Bit(1)
"Auswertung Clear_Data"	ausgeschaltet	Bit(2)
"Diagnose binärer Klemmen"	wird in das PA gemappt	Bit(4)
"Erweiterte PA-Aktualisierung"	gemaess PA-Aktualisierung	Bit(3)
"Prozessabbildaktualisierung"	freilaufend	Bit(6)
"Reaktion bei PROFIBUS-Fehler"	Klemmenbuszyklus wird gestop	BitAre
"Reaktion bei Klemmenbus-Fehler"	Datenaustausch wird verlassen	BitAre
"Max. Laenge der Diagnosedaten"	16 Byte	BitAre

Here, various extended parameters of a DP slave, defined in the GSD-file, are listed. The **Parameters** column shows the name of the parameter. The parameter values entered in **Value** column can be altered by double-click or via the right mouse button. In addition, the **Value range** is specified.

If symbolic names are also specified for the parameters in the GSD-file, the **Symbolic names** option can be activated, so that the values can be displayed with these names. For information, the **Length of user parameters** is also given above the table.

Group assignment of a DP slave

Dialog for group assignment of a DP slave

The dialog shows the 'Userparameter' tab. It has checkboxes for 'Freeze Mode' and 'Sync. Mode', both of which are checked. Below is a table with three columns: Group Membership, Sync. Mode, and Freeze Mode.

Group Membership	Sync. Mode	Freeze Mode
Gr 1	X	X
Gr 2	X	X
Gr 3	X	X
Gr 4	X	X
Gr 5	X	X
Gr 6	X	X
Gr 7	X	X
Gr 8	X	X

At the bottom of the dialog is a button labeled 'Global Group Properties'.

This dialog is used for assigning the slave to one or more of the eight possible groups. The universally applicable group properties (**Sync. Mode** and/or **Freeze Mode**), on the other hand, are defined during configuration of the master's properties (see above 'DP parameters of the DP master, Group properties'). This dialog can also be reached via the **Global Group Properties** button.

The group(s) to which the slave is assigned are marked with a plus sign. The assignment to or removal from a group is accomplished by selecting the group name in the **Group Membership**

column and pressing 'Add slave to group' or 'Remove slave from group' with the right mouse button, or by clicking again with the mouse to the left of the group name.

A slave device can only be assigned to those groups whose properties it supports. The concerned properties of each slave (**Sync. Mode** / **Freeze Mode**) are displayed above the table. The modes supported by the device are checked.

Module parameters of a DP slave

The module parameters dialog of a DP slave matches that of the other modules (see Chapter 6.6.5). The parameters assigned to the slave in addition to the DP and user parameters in the configuration file are displayed here, and the values can be edited in the standard case.

Properties of a DP slave in slave operation of the Profibus

If a Profibus runs in slave mode, the slave device is inserted in the master level of the configuration tree. The configuration can be done in the following dialogs (for a description see the chapters above):

Base parameters

DP parameters

Module parameters

Input/Output

6.6.8 Configuration of CANopen modules

CoDeSys supports a hardware configuration according to CANopen Draft Standard 301. The configuration looks like that described for the hardware dependant configuration.

All EDS (Electronic Data Sheet) respectively DCF (Device Configuration File) files which are stored in the configuration files directory (see Chapter 4.2, Project Options: Directories) when **CoDeSys** is started, can be integrated, edited and displayed in the configuration according to the description in the configuration file *.cfg. In the EDS file the configuration options of a CAN module are described. If you add a module, which is described in a DCF file, only the IEC addresses can be modified, for the module has already been configured completely in a CAN configurator.

The CAN modules get a configuration which describes their time- and error-dependant behaviour during the exchange of messages (dialog 'CAN Parameters of a CAN Module'). Further on for each module the mapping of the PDOs (Process Data Objects), serving for sending and receiving (dialogs 'Receive...' resp. 'Send PDO-Mapping'). The values of the available SDOs (Service Data Objects) can be adapted (dialog 'Service Data Objects').

Additional parameters of a CAN module resp. CAN Master, specified in the device description file can be configured in dialog 'Modulparameters'.

If a CoDeSys-programmable PLC should be integrated in a CAN network as a CANopen slave (also named „CANopen node“ and in the following called „CAN device“), it can be configured in the CoDeSys PLC configurator and the configuration can be stored in form of a EDS-file. This device description file then can be used in any desired CANopen Master configuration.

The following chapters describe the dialogs and menu commands of the CANopen configurator. Further information is provided by the document „CANopen für 3S Laufzeitsysteme V<version>.pdf“.

Base parameters of a CAN Master

The Base parameters dialog of a CAN master matches that of the other modules (see chapter 6.6.5, 'Base parameters of an I/O Module').

CAN parameters of a CAN Master

The properties for transmission on the CAN bus can be set directly after the insertion of the module or can be called up with the command '**Extras**' '**Properties**'.

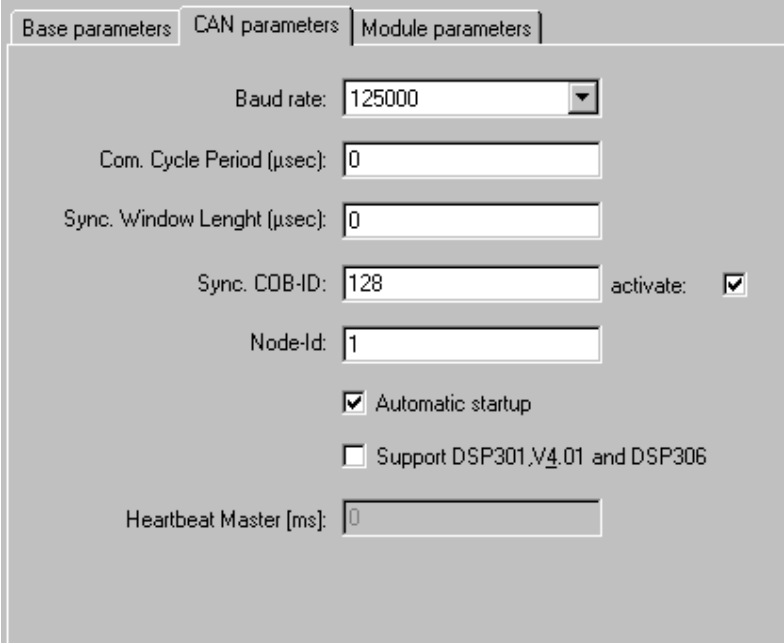
Using the selection option, set the required **Baud rate** which the transmission should take place at.

One differentiates between synchronous and asynchronous transmission modes (see PDO properties) for PDO's (Process Data Object). The synchronisation message is sent with a unique number **Sync. COB-ID** (Communication Object Identifier) in the interval in microseconds which is given by the **Communication Cycle Period**. The synchronous PDO's are transmitted directly after the synchronisation message in the time window (**Sync. Window Length** in microseconds). No synchronisation message will be sent if the fields Comm. Cycle Period and Sync. Window Length contain 0.

activate: Only if this option is activated synchronization messages will be transmitted between master and slaves.

Node-Id: serves to identify the CAN module uniquely and corresponds to the set number on the module itself which is between 1 and 127. The Id must be entered as a decimal number. (Do not mix up with the 'Node number' !)

CAN Parameter Dialog for a CAN-Master



The CAN bus will automatically initialised and started when downloading is occurring and when the controller system starts up if the option **Automatic start** is activated. The CAN bus must be started up in the project if this option is not active.

If the option **Support DSP301,V3.01 and DSP306** is activated, then modular CAN Slaves as well as some additional extensions concerning the standards DSP301 V3.01 and DSP306 will be supported. In this case e.g. the stroke of the Heartbeat will be adjustable (**Heartbeat Master [ms]:**). Working with Heartbeats is an alternative guarding mechanism: In contrast to the Nodeguarding functionality it can be executed by Master- and Slave-Modules. Usually the master will be configured to send heartbeats to the slaves.

Module parameters of a CAN-Master

The module parameters dialog of a CAN master is the same as that for the other modules (see Chapter 6.6.5): The parameters which have been additionally assigned to the master in the configuration file, are displayed here and as a default the values can be edited.

Base parameters of a CAN module

The Base parameters dialog of a DP master matches that of the other modules (see chapter 6.6.5, 'Base parameters of an I/O Module').

IEC addresses by which the PDO's (Process Data Object) in the project can be addressed are entered for **output** and **input addresses**, whereby the direction (input or output) is defined from view of the module.

A marker address must be given at the **diagnostic address** of the CAN module. It works like described for the CAN master.

CAN parameters of a CAN Module

The CAN parameters of a CAN module, which is not acting as master (global watching of the bus), are different to those of a CAN master.

CAN parameters dialog for a CAN Module

Section General:

The **Node-Id** serves to identify the CAN module uniquely and corresponds to the set number on the module itself which is between 1 and 127. The Id must be entered as a decimal number.

If **DCF write** is activated, a DCF file will be created after inserting an EDS file in the defined directory for the compiled files whose name is made up of the name of the EDS file and the Node Id which is tacked on the end.

If the option **Create all SDO's** is activated, then for all objects SDO's will be created (not only for those that have been modified).

If the option **Reset node** (the availability depends on the content of the device description file), is activated, then the slave will be reset before downloading the configuration.

If the option **Optional device** is activated (availability in dialog is target dependent), the master only once will try to read from this node. Then the node, if not answering, will be ignored, i.e. the master will return to normal operation mode.

If the option **No initialization** is activated (availability in dialog is target dependent), the master immediately will activate the node, without sending configuration SDOs. (The SDO data nevertheless will be created and saved on the controller.)

If supported by the target, the creation of SDOs can be suppressed in three levels. This might be desired in case of having limited space for configuration:

Attention: These settings should not be modified without having detailed knowledge of the system!

CreateCommSDOs: Communication parameter SDOs

CreateMappingSDOs: Mapping configuration SDOs

CreateBasicSDOs: Base parameters SDOS (Nodeguarding, Sync etc.)

Only for the activated types the appropriate SDOs will be created. The above described option 'Create all SDO's' only refers to currently here activated SDO types.

Section Node guard: (alternatively to guarding by the Heartbeat mechanism)

If the option **Nodeguarding** is activated, a message will be sent to the module according to the interval set by **Guard Time** in milliseconds. If the module does not then send a message with the given **Guard COB-ID** (Communication Object Identifier), it will receive the status "timeout". As soon as the number of attempts (**Life Time Factor**) has been reached, the module will receive the status "not OK". The status of the module will be stored at the diagnosis address. No monitoring of the module will occur if the variables Guard Time and Life Time Factor are not defined (0).

Section Heartbeat Settings: (alternatively to Nodeguarding)

If the option **Activate Heartbeat Producer** is activated, the module will send heartbeats according to the interval defined in **Heartbeat Producer Time:** given in **ms**.

If the option **Activate Heartbeat Consumer** is activated, then the module will listen to heartbeats which are sent by the master. As soon as no more heartbeats are received, the module will switch off the I/Os.

Section Emergency Telegram:

A module sends an **emergency** message, with a unique **COB-Id.**, when there is an internal error. These messages, which vary from module to module, are stored in the diagnosis address.

The entries "FileInfo" and "DeviceInfo" of the EDS or DCF file from the corresponding module manufacturer are hidden behind the **Info** button.

CAN Modules Selection for Modular Slaves

In the left column (**Available modules**) you find all modules which are available for the slave. Mark the desired modules and by using the buttons **Add** and **Remove** create a selection in the right column (**Selected Modules**). The PDO- and SDO selection will be updated automatically.

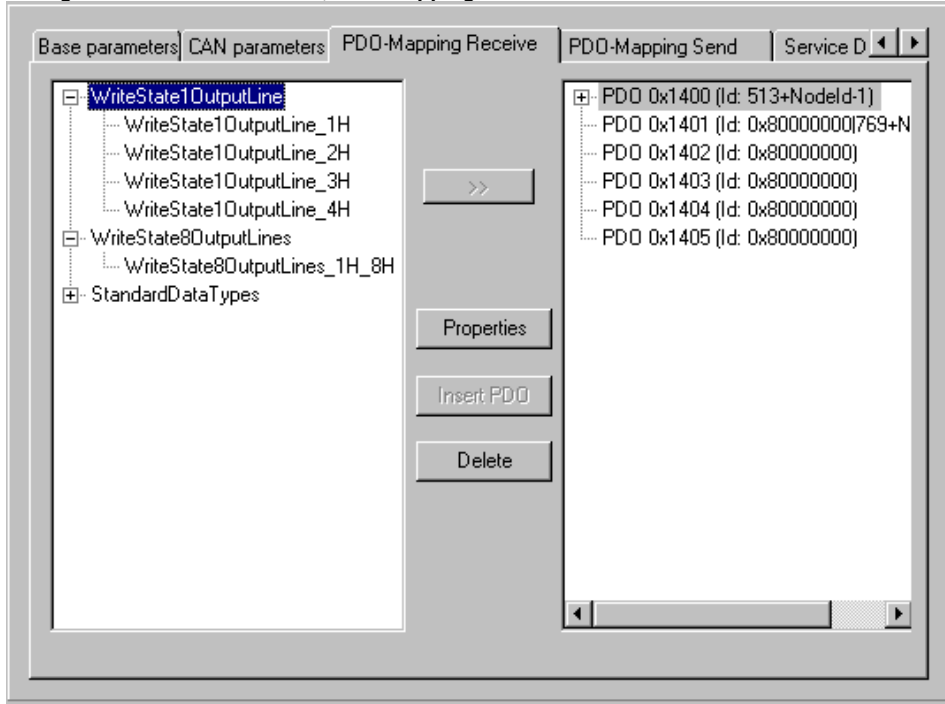
PDO mapping of a CAN module

The tabs **Receive PDO mapping** and **Send PDO mapping** in the configuration dialog for a CAN module allow the "mapping" of the module, which is described in the EDS file, to be changed.

All of the "mapable" objects in the EDS file are located on the left side and can be added in the right side to the PDO's (Process Data Object) (">>" button) or removed again (**Remove** button). The StandardDataTypes can be inserted to create empty spaces in the PDO.

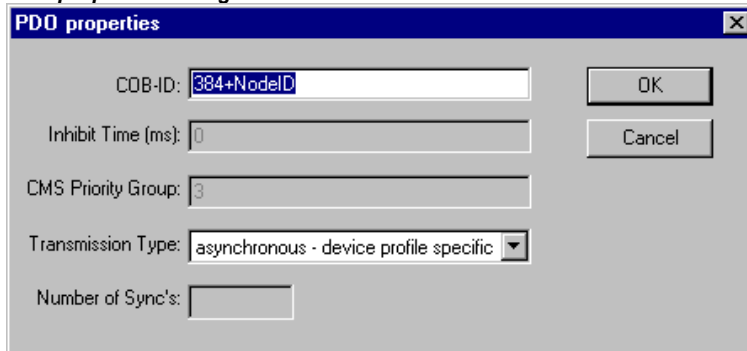
The button **Insert Element** can be used to create further PDO's and to add appropriate objects to them. The allocation of inputs or outputs to the IEC addresses can be achieved over the inserted PDO's. The settings which have been made in the controller system configuration will become visible when one leaves the dialog. The individual objects can be afforded symbolic names there.

Configuration of a CAN module, PDO-Mapping



The standard set properties of the PDO's can be edited using command **Properties**.

PDO properties dialog



Each PDO message requires a unique **COB-Id** (Communication Object Identifier).

The field appears in grey and cannot be edited if an option is not supported by the module or if the value cannot be changed.

The **Inhibit Time** is the minimum time between two messages from this PDO. This is to prevent PDO's which are sent when the value is changed from being sent too often.

The **CMS Priority Group** cannot be changed and describes the relative importance of the PDOs during the CAN transmission. Values from 0 to 7 are displayed, whereby 0 is the highest.

Transmission Type offers you a selection of possible transmission modes for this module:

acyclic - synchronous: the PDO will be transmitted synchronously but not periodically.

cyclic – synchronous: the PDO will be transmitted synchronously, whereby the **Number of Sync's** gives the number of synchronisation messages, which lie between two transmissions of this PDO.

synchronous – RTR only: the PDO will be updated after each synchronisation message but not sent. It is only sent when there is an explicit request to do so (**Remote Transmission Request**)

asynchronous – RTR only: the PDO will only be updated and transmitted when there is an explicit request to do so (**Remote Transmission Request**)

asynchronous – device profile specific and **asynchronous - manufacturer specific:** the PDO will only be transmitted when specific events occur.

Number of Syncs: If cyclic transmission has been set, enter here the number of synchronisation messages (see 'Com. Cycle period' in the CAN parameter dialog) which should be sent between two transmissions of the PDO.

Event-Time: If an corresponding transmission type is set, enter here in milliseconds (ms) the interval between two transmissions.

Service Data Objects

Here you find a list of all objects in the EDS or DCF file which are in the area of the Index 0x2000 to 0x9FFF and which are marked as writable.

The properties **Index**, **Name**, **Value**, **Type** and **Default** are displayed for every object. The value can be changed. Mark the value and press the <Space bar>. After making the change confirm the new value with <Enter> or reject it with the <Escape> key.

The set values are transmitted in the form of SDO's (Service Data Object) to the CAN modules at the initialisation of the CAN bus.

Resetting a SDO to its default value: If a SDO, after it has been transmitted to the CAN bus with a value modified by the user (i.e. with a value deviating from the default value), should be reset to its default value for the next transmission, delete the entry in the 'Value' column. This means, the value field must be **empty** for this purpose; if you enter the default value, nothing will be transmitted!

Note: All incompatible data types between CANopen and IEC-61131 will be replaced in **CoDeSys** by the next larger IEC-61131 data type.

Dialog for configuration of the Service Data Objects (SDO)

Index	Name	Value	Type
2100sub1	8bit Output Block No. 1		Unsigned
2100sub2	8bit Output Block No. 2		Unsigned
2100sub3	8bit Output Block No. 3		Unsigned
2100sub4	8bit Output Block No. 4		Unsigned
2100sub5	8bit Output Block No. 5		Unsigned
2100sub6	8bit Output Block No. 6		Unsigned
2100sub7	8bit Output Block No. 7		Unsigned
2100sub8	8bit Output Block No. 8		Unsigned
2100sub9	8bit Output Block No. 9		Unsigned

6.6.9 Configuration of a CanDevice (CANopen Slave)

A PLC which is programmable with CoDeSys can be used as a CANopen Slave (CANopen-Node, called "CanDevice" in the following) in a CAN network.

For this purpose the PLC can be configured in the CoDeSys PLC Configurator and the configuration can be saved in an EDS-file. This EDS-file (device file) later can be used in any CANopen Master configuration.

Preconditions for creating a CanDevice in the CoDeSys PLC Configurator:

1. The libraries

3S_CanDrv.lib

3S_CanOpenManager.lib

3S_CanOpenDevice.lib

must be included in the CoDeSys project. They are needed for running the PLC as an CANopen device.

2. In the configuration file (*.cfg) on which the configuration is basing, an appropriate entry for a CanDevice must be inserted. Only then in the PLC Configuration Editor a subelement 'CanDevice' can

be appended and parameterized in the three configuration dialogs which will be described in the following:

- Base settings
- CAN settings
- Default PDO mapping

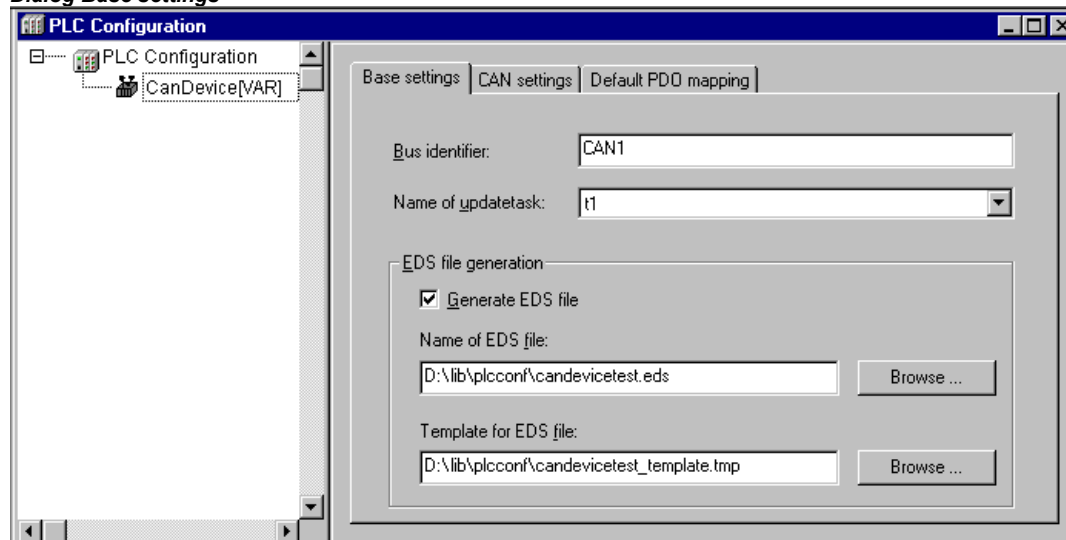
Base settings of a CanDevice

Bus identifier: currently not used.

Name of updatetask: Name of the task, in which the CanDevice is called. A selection list will provide all tasks which are available in the project.

EDS file generation: Activate this option if you want to generate a device file (EDS file) from the current configuration settings in order to be able to use the CanDevice later in any master configuration. Enter a path and name for the file in the field **Name of EDS file**. Optionally a manually created template file can be defined (**Template for EDS file**), which will be supplemented with the settings done in the configuration dialogs. For example you could create a text file containing certain EDS file entries, save it as "EDS_template.txt" and enter the path of this template in the current dialog. If you then generate an EDS file "device_xy.eds" from the current project, the entries resulting from the project will be merged with those of the template and will be saved in "device_xy.eds". (Do not use the extension ".eds" for the template file !) If entries are created by the current project which are already defined by the template, the template definitions will not be overwritten.

Dialog Base settings



For entering the file paths you can use the standard dialog for browsing for a file which can be opened by using the button **Browse....**

CAN settings of a CanDevice

Here you can specify CAN parameters as known from the CANopen standard in order to give them to the runtime system. It depends on the runtime system how they will be interpreted. In case you are using the CANopen implementation realized in the 3S runtime systems via the library CanOpen.lib, please see the corresponding documentation "*CanOpen for 3S Runtime systems*".

Node Id: The Node Id (1-127) is the node number which is used by the master for addressing the device in a CANopen network.

Baud rate: From the selection list choose the desired baud rate for the transmission within the bus.

Device Type: In this field automatically the type of the device is displayed, which is returned when looking for object 0x1000, i.e. the type of the PLC which is configured in the project. The device type can be edited.

Automatic startup: If this option is activated, at a download or at starting up the PLC the CAN bus will be initialized and started automatically: If the option is not activated, the CanDevice will wait for an appropriate command.

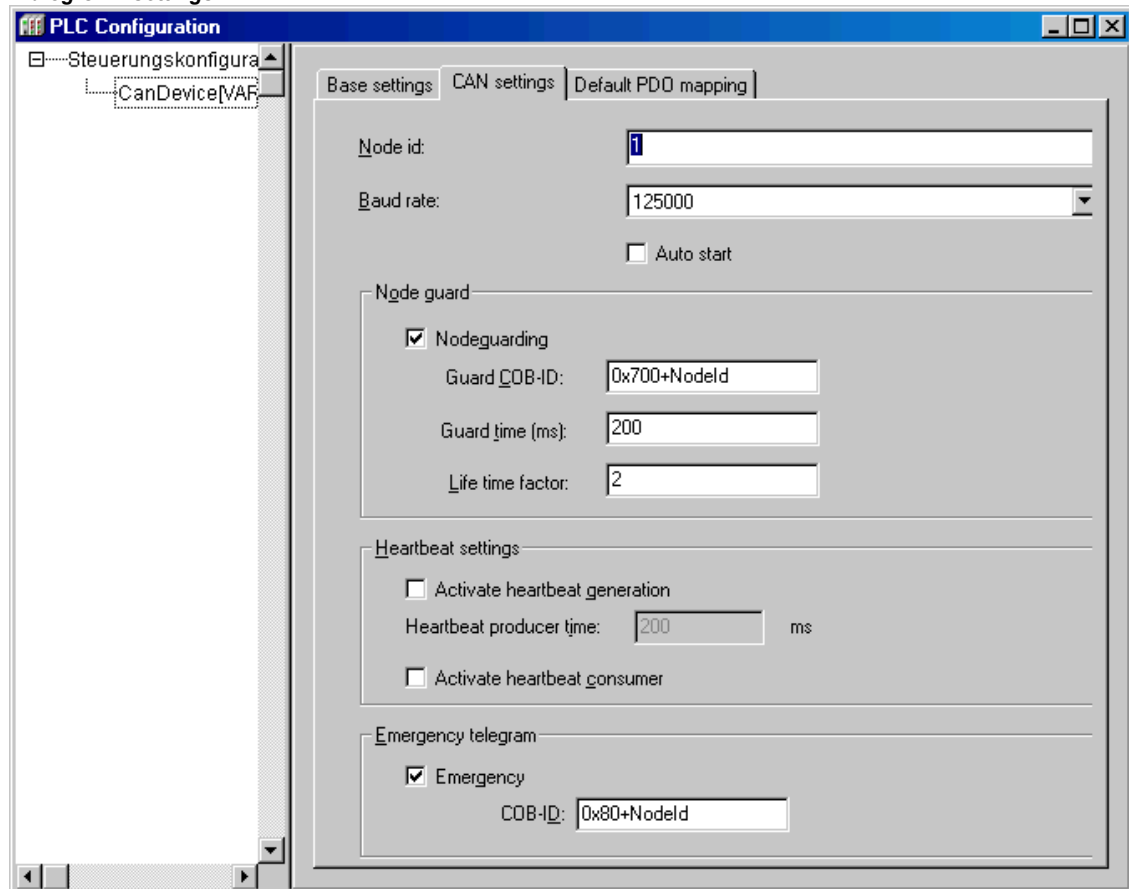
A configuration of **Nodeguard-** and **Emergency Telegram** functionality is possible. Please refer to the corresponding description for the configuration of CAN Modules and Master.

Heartbeat settings:

If option **Activate heartbeat generation** is activated, the CanDevice will send heartbeats according to the interval (milliseconds), defined at **Heartbeat Producer Time** .

If option **Activate heartbeat consumer** is activated, the CanDevice will listen to heartbeats, which are sent by the module defined by **Consumer ID**. The **Heartbeat Consumer Time** defines the time (in milliseconds) after which an error flag will be set in case no heartbeat has been received.

Dialog CAN settings



Default PDO mapping of a CanDevice

In this dialog the entries of the local Parameter Manager can be assigned to PDOs, which will be sent/received by the CanDevice. The PDOs then will be available for the PDO mapping in any master configuration where the CanDevice is integrated.

In the Parameter Manager lists the parameter entries are linked to project variables via index/subindex.

Please regard: Subindex 0 of an index, which implies more than one subindex, will be used implicitly for storing the number of subindices. For this reason do not use subindex 0 in the Parameter Manager. Also regard that the parameters of a particular index must be entered in ascending order (subindices 1,2,3...) in the Parameter Manager.

List of mapable objects: Choose from the selection list the variables' parameter list, for whose entries the CanDevice should generate PDOs. If supported by the target system, parameter lists of type 'Mapping' can be created in the Parameter Manager, which contain process variables especially

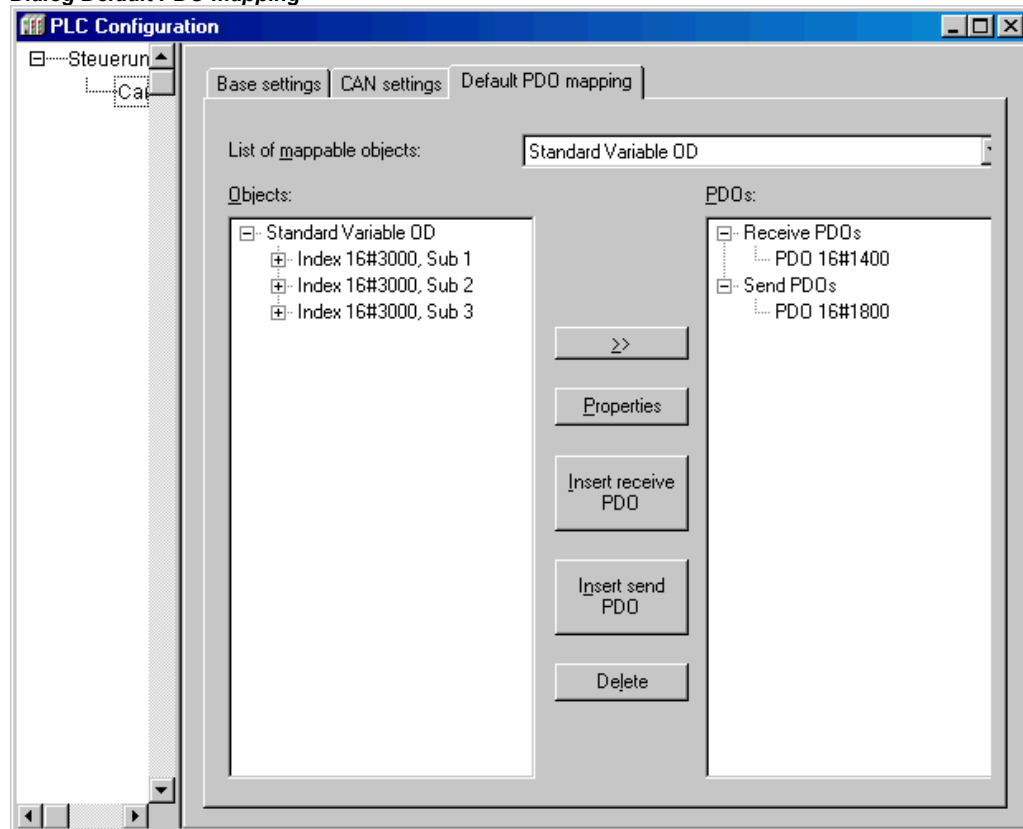
intended for the PDO mapping of a CANDevice. In this case only these parameter lists will be offered here in the mapping dialog. Otherwise all available parameter lists of type 'Variables' and 'Instance' will be offered.

!Attention: If in the target settings for the Parameter Manager an "Index range for mappings" is defined, then the CanDevice will **only** regard that range for the mapping and no other index ranges possible also defined!

According to the chosen parameter list the **Objects** will appear in the left window. In the right window you create the desired PDO configuration (**PDO's**). Via the buttons **Insert receive PDO** resp. **Insert send PDO** there you can insert 'Receive PDOs' and 'Send PDOs' below the corresponding list organizing elements. In order to assign an object of the left window to one of these send or receive PDOs, mark the object in the left window and also the PDO in the right window and then press **>>**. Thereupon the object will be inserted below the PDO in the right window. The **Properties** of the PDO can be defined in a dialog which is also used for the PDO configuration of other CAN modules.

By using button **Delete** the PDO currently marked in the right window will be removed from the configuration.

Dialog Default PDO mapping



Example:

objective: On the first Receive PDO (COB-Id = 512 + Nodeld) of the CanDevice the variable PLC_PRG.a should be received.

Thus in the Parameter Manager in a variable list a index/subIndex must be assigned to variable PLC_PRG.a. The Parameter Manager can only be opened, if it is activated in the target settings in category 'Network functionality' and if valid index and subindex ranges are defined there.

Now in the dialog 'Default PDO-Mapping' of the CanDevice the index/subindex entry of the respective parameter list can be assigned to a Receive PDO.

6.6.10 Configuration of DeviceNet Modules...

CoDeSys supports a hardware configuration for a bus system, which is using the internationally standardized DeviceNet protocol (EN50325). DeviceNet mainly is used to realize Master-Slave networks having Plug & Play properties, thus having a bus for direct connection to sensors and actuators (proximity switches, outlets).

The DeviceNet communication protocol bases on CAN (Controller Area Network). A direct connection between the communing modules is a precondition for the data exchange.

The CoDeSys DeviceNet configuration editor provides for the definition of a DeviceNet-Master which will control the data exchange within the network. Various communication types are supported for the exchange of the in- and output data between the slave modules (DeviceNet-Slave) . Usually the DeviceNet-Master takes the "UCMM"-function (Unconnected Message Manager for multiple connections) and takes care of requests from other masters to its slaves.

A configuration file *.cfg allowing the insertion of DeviceNet-Master and Slave modules is a precondition for doing a DeviceNet configuration in the CoDeSys PLC Configuration. The configuration file automatically is searched in the currently set directory for configuration files (see Chapter 4.2, Project Options: Directories).

According to the definitions in the configuration file *.cfg the EDS files) (Device description files, Electronic Data Sheet), which are also found in the currently set configuration files directory, can be used for the configuration. In the EDS file the configuration options of a DeviceNet module are described. Regard that CAN device description files also have the extension ".EDS", but are not usable for a DeviceNet configuration!

Generally regard the possibility of adding configuration files during working on the project (see Chapter 6.6.2., Add configuration file).

If a DeviceNet-Master is selected in the configuration tree, the following dialogs are available on appropriately named tabs: Base parameters, DeviceNet Parameters, Module parameters.

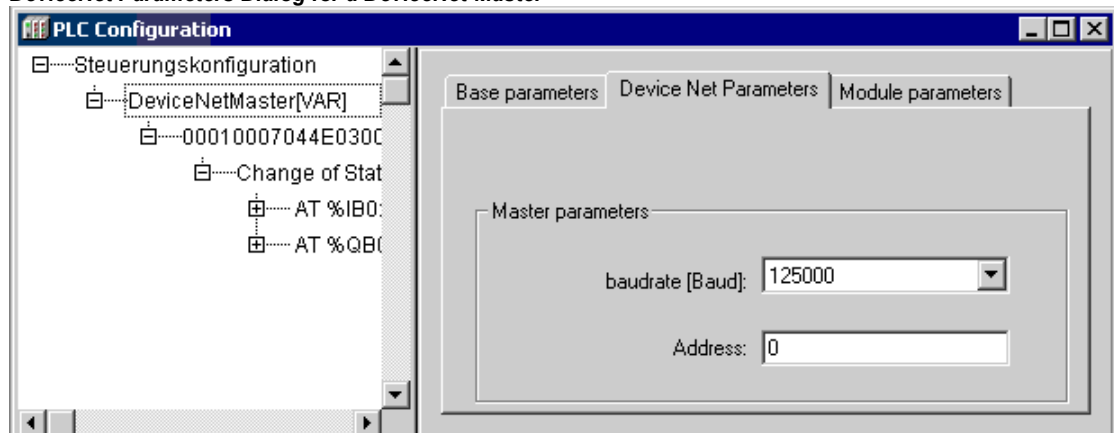
If a DeviceNet-Slave is selected which is inserted below a DeviceNet-Master, the following dialogs will be available: Base parameters, DeviceNet parameters, I/O connection configuration, Parameters, Module parameters.

Base parameters of a DeviceNet-Master

The , Base parameters dialog of a DeviceNet-Master referring to the items **Module id**, **Node number**, **Input address**, **Output address** and **Diagnosis address** matches that of the other modules (see 'Chapter 6.6.5, Base parameters of an I/O Module).

DeviceNet Parameters of a DeviceNet-Master

DeviceNet Parameters Dialog for a DeviceNet-Master



In field **Address** insert the DeviceNet-Master identification number, which is set at the module itself. The meaning of this ID corresponds to that of the "Node-ID" of a CAN module and must not be mixed up with the Node number, or the Address defined in the Base parameters dialog!. It must be entered decimal, possible values: 0-63, default setting:0.

Also the **Baudrate [Baud]** for the data exchange within the network is defined here. Choose one of following settings: 125000 (default), 250000, 500000.

Module Parameters of a DeviceNet-Master

The module parameters dialog of a DeviceNet-Master is the same as that for the other modules (see Chapter 6.6.5, Module parameters of an I/O Module). The parameters which have been additionally assigned to the master in the configuration file, are displayed here and usually the values can be edited.

Base parameters of a DeviceNet-Slave

The base parameters dialog of a DeviceNet-Slave referring to the items **Input address** and **Output address** matches that of the other modules (see Chapter 6.6.5, Base parameters of an I/O Module). The direction (input or output) is defined from view of the module.

DeviceNet Parameters of a DeviceNet-Slave

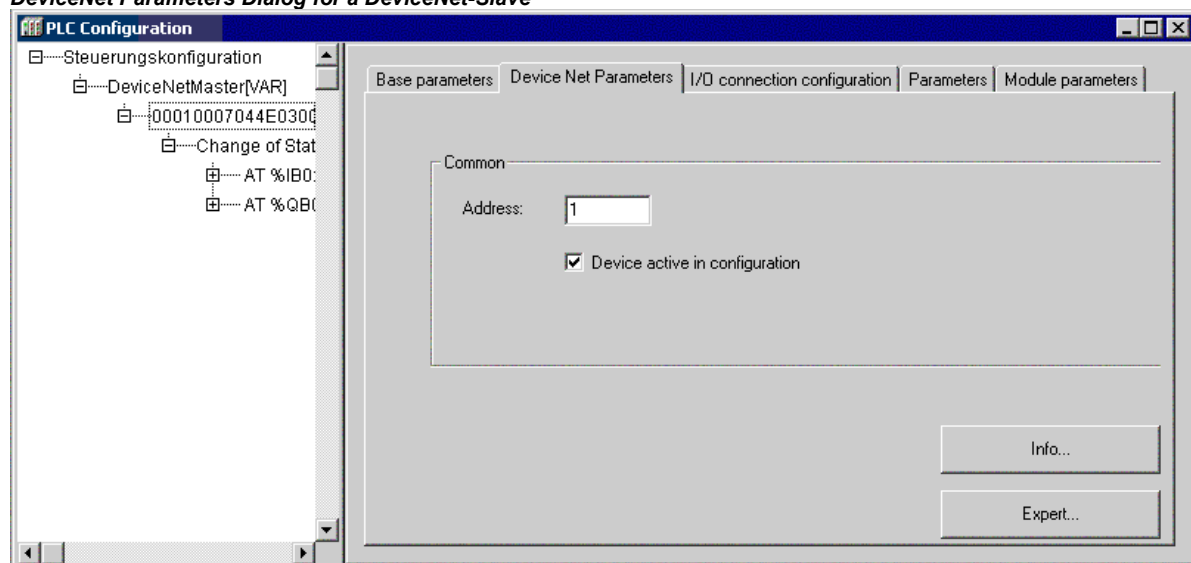
Here the common parameters of the module are configured:

Address: Identification of the DeviceNet-Slave, set at the module itself. The meaning of this ID corresponds to that of the "Node-ID" of a CAN module and must not be mixed up with the 'Node number, or the "Address" defined in the Base parameters dialog!). It must be entered decimal, possible values: 0-63, default setting:0.

Device active in configuration: Activate this option in order to make the device an active participant concerning the data exchange in the network.

Info...: This button opens a window displaying the content of the EDS file. Please regard, that CAN device description files also have the extension ".EDS", but are not usable for a DeviceNet configuration!

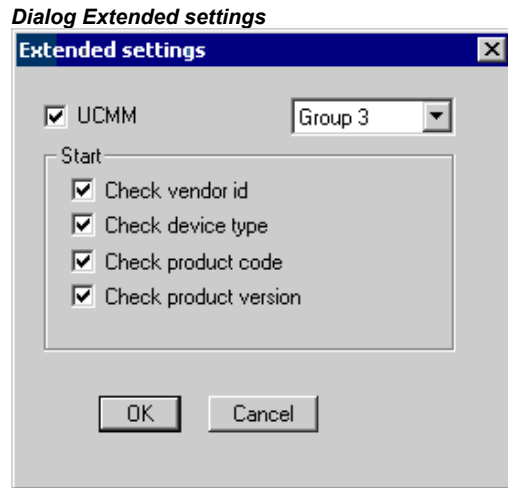
DeviceNet Parameters Dialog for a DeviceNet-Slave



Expert...: This button opens the dialog **Extended settings**, where the following settings can be changed:

UCMM: (Unconnected Message Manager for multiple connections) If this option is activated (default), the slave is able to handle UCMM messages. The possible classifications: **Group1**, **Group2** or **Group3** (default)

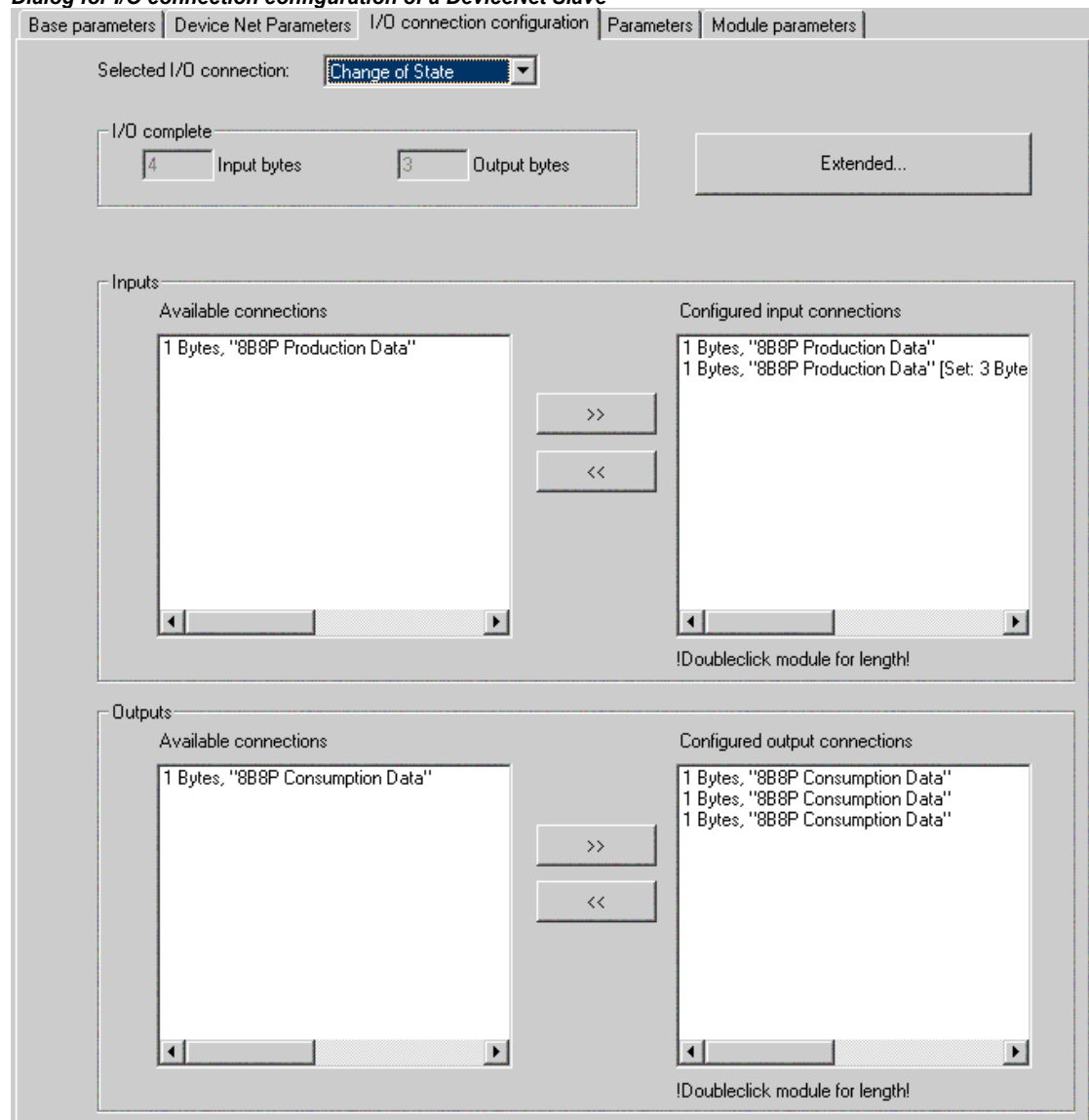
The following checks will be done per default at the **Start** of the network and may be deactivated here, if necessary. During the check always the value given by the EDS file will be compared to that found at the device: **Check vendor id**, **Check device type**, **Check product code**, **Check product version**



I/O connection configuration of a DeviceNet-Slave

Here you configure the inputs and outputs of the slave, via which the data (parameter values) should be exchanged. A connection type is to be defined and a selection of inputs and outputs is to be composed according to the possibilities given by the module (EDS file, Inputs, Outputs).

Dialog for I/O connection configuration of a DeviceNet Slave



Selected I/O connection: Select one of the following connection types, which should be valid for the I/O connection defined below:

Poll: The data of the slave will be polled cyclically (Master-Slave-process)

Bit Strobe: The DeviceNet-Master sends a broadcast telegram to all slaves requesting them to send the current data. The slaves will answer one after the other, starting with node 1.

Change of State: The slave will send its data to the master at each change detected at an input. No explicit request by the master is needed.

Cyclic: The slave will send its data after a defined cycle time without an explicit request by the master ("Heartbeat" function).

Multicast Poll: currently not supported

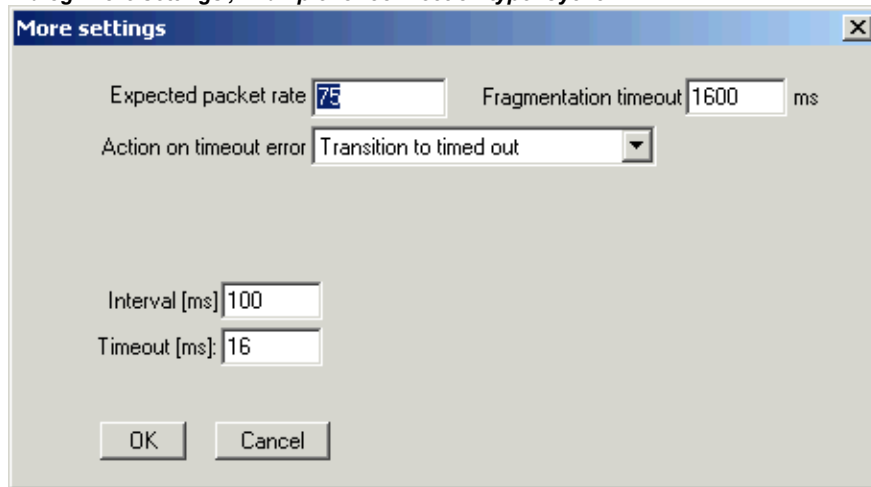
I/O complete: Here the sums of **Inputbytes** and **Outputbytes**, currently used for all configured inputs and outputs, will be displayed. The sums are calculated from the lengths defined for the I/Os in the 'Inputs' and 'Output' areas of the dialog.

Extended: This button opens the dialog **More settings**, which allows to modify the following default settings for the currently chosen connection type:

Expected Packet Rate: Default: 75, expected rate (in milliseconds) according to which the slave is expected to send its data over the current connection.

Fragmentation timeout: [ms]: Default 1600 ms; If the data to be sent are exceeding a size of 8 Bytes, they must be fragmented and send in several telegram packages. The fragmentation timeout defines in milliseconds, how long the master should wait for an answer of the slave on a fragmented telegram, before triggering the action defined in 'Action on timeout error'.

Dialog 'More settings', Example for connection type 'Cyclic'



Action on timeout error: Define, which of the following actions should be started in case of a timeout:

Transition to timed out: (Default) This action is defined slave-specifically.

Auto delete: The I/O connection will be deleted.

Auto reset: The connection persists, the master re-configures the slave, the watchdog is reset.

Further options for connection type 'Change of state':

Lock time for sending: (Default:1) Minimum interval (in milliseconds) between two messages, even if data have changed before this time span is over. This method helps to avoid overloading the device with incoming requests. "0" means no lock time, in this case the data will be exchanged as fast as possible.

Timeout[ms]: (Default: 16) If the heartbeat rate has been exceeded by this time span (in milliseconds) without data having been sent, a timeout error will be detected.

Heartbeatrate[ms]: (Default: 250) Time span in milliseconds, after which the slave in any case must send its data, even if they have not changed.

Further options for connection type 'Bit Strobe':

Use output bit: When answering to the master, the slave will use that output bit, which corresponds to the output bit used by the master in the request telegram.

Further options for connection type Cyclic':

Interval [ms]: Time interval in milliseconds, according to which the slaves automatically has to send its data (Heartbeat).

Timeout [ms]: If the heartbeat rate has been exceeded by this time span (in milliseconds) without data having been sent, a timeout error will be detected.

Inputs:

From field **Available connections** select the desired inputs and transfer them to field **Configured input connections** by button >> . By button << you can remove entries from there.

In order to modify the length of a configured input, perform a double-click on this entry. The dialog **Length of connection** will open. Enter here the desired **Length in Bytes** and confirm with OK. The length thereupon will be displayed in brackets behind the configured input.

Configured inputs will be visible immediately in the configuration tree. Indented below the slave entry there will be an entry with the name of the connection type. Below that the respective inputs and outputs will be inserted.

Outputs:

Configure the outputs like described for the inputs.

Parameters of a DeviceNet-Slave

The parameters listed here are given by the EDS file. According to the defined I/O connection configuration their current values will be exchanged in the network.

Obj.: Identification of the parameter (object), which is used to access the parameter in a parameter list (object dictionary). This object number is created from the parameter number given by the corresponding parameter description in the EDS file (section [Params], "Param<number>").

Typ: Data type of the parameter

Acc.: Access rights: rw=read and write, ro=read only

Min., Max.: Value range of the parameter, limited by the minimum and maximum value

Default: Default value of the parameter

Value: As it is defined in the EDS file, the parameter value might be edited here. Either a selection list of permissible values is available or an edit field can be opened by a mouse-click on the table cell.

Module parameters of a DeviceNet-Slave

The module parameters dialog of a DeviceNet-Slave is the same as that for the other modules (see Chapter 6.6.5, Module parameters of an I/O Module): The parameters which have been additionally assigned to the master in the configuration file, are displayed here and as a default the values can be edited.

6.6.11 PLC Configuration in Online Mode

In online mode the PLC configuration displays the states of the inputs and outputs of the PLC. If a boolean input or output has the value TRUE, the little box at the beginning of the entry line in the configuration tree will get blue, non-boolean values will be added at the end of the entry line (e.g. "=12").

The boolean inputs can be toggled by mouse-clicks. At other inputs a mouse-click on the beginning of the line opens a dialog, where the value can be modified. The modified value will be set in the PLC as soon as the dialog is closed with **OK**.

Also regard the target specific possibilities for online diagnosis.

6.6.12 Hardware scan/State/Diagnosis information from the PLC

If supported by the target system and the actual configuration file (*.cfg), information on the structure, the status and diagnosis results of the currently connected hardware can be get from the PLC and displayed in the PLC Configuration in CoDeSys:

Scan module configuration

If supported by the target system and the actual configuration file (*.cfg), the command **Scan module configuration** will be available in the context menu for the module which is currently selected in the PLC Configuration tree.

This command is only available in offline mode. If it is activated, the actual hardware configuration of the particular module on the PLC will be scanned and automatically be offered for inserting in the configuration tree of the CoDeSys PLC Configuration. Thus the existing module configuration can easily be mapped in CoDeSys.

Load module state

If supported by the target system and the actual configuration file (*.cfg), the command **Load module state** will be available in the context menu for the module which is currently selected in the PLC Configuration tree.

This command is only available in online mode. If it is activated, the actual status of the module will be read from the PLC and get displayed by a special color in the configuration tree:

Black: Module existing and parameterized correctly.

Blue: Module existing but parameterized incorrectly.

Red: Module not found.

An update of the status display also automatically will be done at each download.

Show diagnosis messages

If supported by the target system and the actual configuration file (*.cfg), the command **Show diagnosis messages** will be available in the context menu for the module which is currently selected in the PLC Configuration tree. This command is only available in online mode. If it is activated, actual diagnosis messages for the module coming from the PLC will be displayed in a CoDeSys window.

6.7 Task Configuration

6.7.1 Overview

In addition to declaring the special PLC_PRG program, you can also control the processing of your project using the task management.

A **Task** is a time unit in the processing of an IEC program. It is defined by a name, a priority and by a type determining which condition will trigger the start of the task. This condition can be defined by a time (cyclic, freewheeling) or by an internal or external event which will trigger the task; e.g. the rising edge of a global project variable or an interrupt event of the controller.

For each task you can specify a series of programs that will be started by the task. If the task is executed in the present cycle, then these programs will be processed for the length of one cycle.

The combination of priority and condition will determine in which chronological order the tasks will be executed.

Note: Please do not use the same string function (see standard.lib) in several tasks, because this may cause program faults by overwriting.

Each task can be enabled or disabled explicitly.

For each task you can configure a watch dog_(time control) can be configured; the possible settings depend on the target system.

In the Online Mode the task processing can be monitored in a diagram.

Additionally there is the possibility to link **System events** (e.g. Start, Stop, Reset) directly with the execution of a project POU.

The

the target file. If the standard descriptions are extended by customer specific definitions, then those will be displayed in an additional tab 'Parameter' in the right part of the window.

6.7.2 Working in the Task Configuration

The most important commands you find in the context menu (right mouse button).

At the heading of the Task Configuration are the words "Task Configuration." If a plus sign is located before the words, then the sequence list is closed. By double-clicking on the list or pressing <Enter>, you can open the list. A minus sign now appears. By double-clicking once more, you can close the list again. For every task, there is a list of program call-ups attached. Likewise, you can open and close this list the same way.

With the 'Insert' 'Insert Task' command, you can insert a task.

With the 'Insert' 'Append Task' command, you can insert a task at the end of the configuration tree.

With the 'Insert' 'Insert Program Call', a program call will be assigned to the task which is actually selected in the configuration tree.

Further on for each entry in the configuration tree an appropriate configuration dialog will appear in the right part of the window. There options can be activated/deactivated resp. inputs to editor fields can be made. Depending on which entry is selected in the configuration tree, there will be the dialog for defining the 'Taskattributes' (see 'Insert Task'), the dialog for defining a 'Program Call' (see 'Insert Program Call') or the table of 'System events'. The configuration options depend on the target system. The settings made in the dialogs will be taken over to the configuration tree as soon as the focus is set to the tree again.

A task name or program name can also get edited in the configuration tree. For this perform a mouse click on the name or select the entry and press the <Space> button to open an edit frame.

You can use the arrow keys to select the previous or next entry in the configuration tree.

By clicking on the task or program name, or by pressing the <Space bar>, you can set an edit control box around the name. Then you can change the designation directly in the task editor.

'Insert' 'Insert Task' or 'Insert' 'Append Task'

With this command you can insert a new task into the Task Configuration. The entries each consist of a symbol and the task name.

If a task or the entry 'System events' is selected, then the '**Insert Task**' command will be at your disposal. The new task will be inserted after the selected one. If the entry 'Task Configuration' is selected, then the '**Append Task**' is available, and the new task will be appended to the end of the existing list. The maximum number of tasks is defined by the target system. Please regard that a certain number of tasks already might be reserved for modules of the PLC Configuration (defined in the cfg-file).

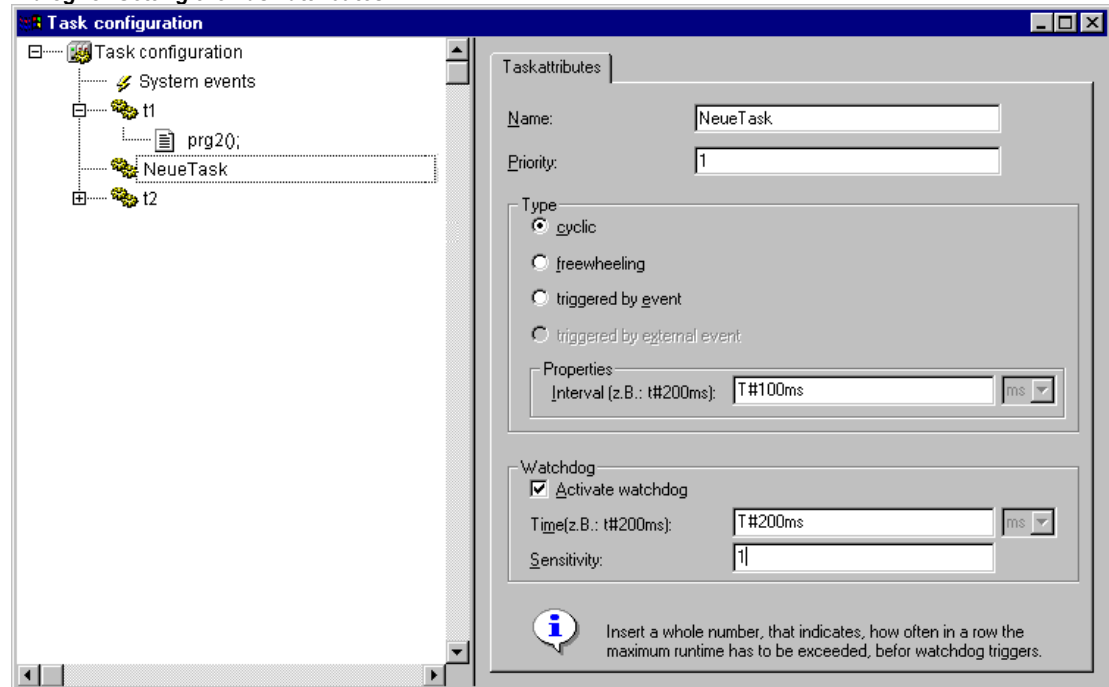
When inserting a task, the dialog for setting the **Task attributes** will be opened.

Insert the desired attributes:

Name: a name for the task; with this name the task is represented in the configuration tree; the name can be edited there after a mouse click on the entry or after pressing the <Space> key when the entry is selected.

Priority (0-31): (a number between 0 and 31; 0 is the highest priority, 31 is the lowest),

Dialog for Setting the Task attributes



Type:

cyclic (

If there is no entry in both of the fields 'Interval' and 'Event', then the task interval will depend on which runtime system is used (see runtime documentation); e.g. in this case for CoDeSys SP NT V2.2 and higher an interval of 10 ms will be used).

Watchdog

For each task a time control (watchdog) can be configured. If the target system supports an extended watchdog configuration, possibly there are predefined upper and lower limits and a default for the watchdog time are defined, as well as a time definition in percent.

Watchdog: When this option is activated (☑) then, regarding the currently set sensitivity (see below), the task will be terminated in error status as soon as the processing takes longer than defined in the 'Time' field (see below).

Attention: Target system CoDeSys SP 32 Bit Full switches off the watchdog function when the flow control is active or when the execution currently is stopped at a breakpoint.

Time (e.g.: t#200ms): Watchdog time; after the expiration of this term, regarding the currently set sensitivity (see below), the watchdog will be activated unless the task has not been terminated already. Depending on the target system the time has to be entered as percent of the task interval. In this case the unit selection box is greyed and shows "%".

Sensitivity: Here you can enter in integer numbers at which overrun of the watchdog time an error should be generated. The default entry is "1", i.e. at the first overrun of the watchdog time an error occurs. Attention: If "0" is entered, the watchdog will be deactivated!

Manufacturer specific attributes:

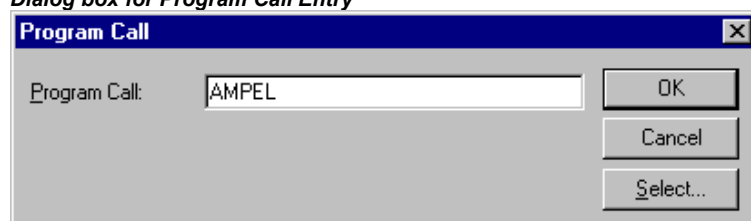
Additionally to these standard attributes for the currently selected task manufacturer specific attributes might be displayed in a second tab "**Parameters**". This will be the case if it is defined in the target-specific description file for the task configuration.

'Insert' 'Insert Program Call' or 'Insert' 'Append Program Call'

With these commands you will open the dialog box for entering a program call to a task in the Task Configuration. Each entry in the task configuration tree consists of a symbol (☐) and the program name.

With '**Insert Program Call**', the new program call is inserted before the selected program call, and with '**Append Program Call**' the program call is appended to the end of the existing list or program calls.

Dialog box for Program Call Entry



In the field 'program call' specify a valid program name out of your project or open the Input Assistant with the **Select** button to select a valid program name. The program name later also can be modified in the configuration tree. For this select the entry and press the <Space> key or just perform a mouse click to open an editor field. If the selected program requires input variables, then enter these in their usual form and of the declared type (for example, prg(invar:=17)).

The processing of the program calls later in online mode will be done according to their order (top down) in the task editor..

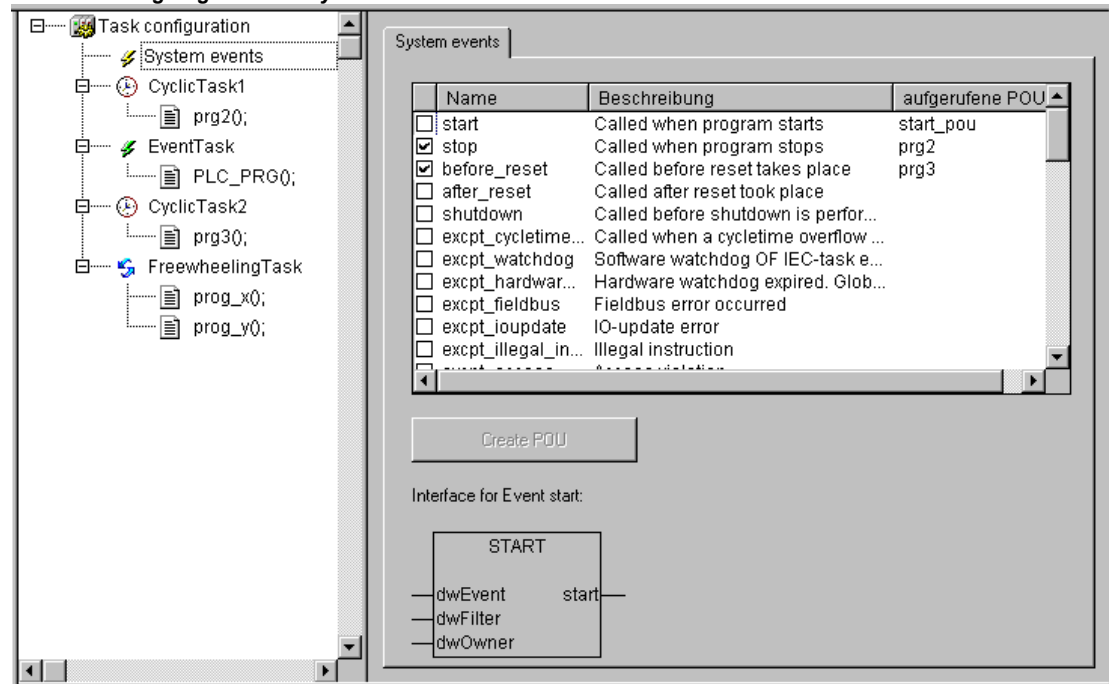
Please regard: Do not use the same string function in several tasks (see Standard Library Elements), because in this case values might be over stroke during processing of the tasks.

6.7.3 System Events

Instead of a "task" also a "system event" can be used to call a POU of your project. The available system events are target specific (definition in target file). The list of the standard events of the target may be extended by customer specific events. Possible events are for instance: Stop, Start, Online Change.

The assignment of system events to POUs is also done in the Task configuration editor. Use the dialog 'Events', which will be opened as soon as the entry "System-events" is selected in the task configuration tree:

Table for Assigning POUs to System Events



Each event is represented in a line: **Name** and **Description** are displayed as defined in the target file, in the column **called POU** you can enter the name of the project POU which should be called and processed as soon as the event occurs.

For this use the input assistant (<F2>) or enter manually the name of an already existing POU (e.g. "PLC_PRG" or "PRG.ACT1"), or insert a name for a not yet existing POU. **Attention for RISC and Motorola 68K target systems:** The name of a function assigned to a system event (callback function) **must** start with „callback“!. In order to get a new POU (function) created in the project, press button **Create POU <name>**. Hereupon the POU will be inserted in the Object Organizer. The input and output parameters which are required by the event will automatically be defined in the declaration part of the POU. Below the assignment table the currently selected event is displayed in a picture, showing the required parameters.

If you actually want the POU to be called by the event, activate the entry in the assignment table (☑). **Activating/deactivating** is done by a mouse click on the control box.

6.7.4 Task Configuration in Online Mode

In online mode the status and number of passed through cycles of each task will be displayed in the configuration tree. The time flow is monitored in a diagram. Precondition: the libraries **SysTaskInfo.lib** and **SysLibTime.lib** must be included in the project to provide functions for the internal evaluation of the task times. The libraries will be included automatically as soon as a target is set which supports the task monitoring.

Display of task status in the configuration tree:

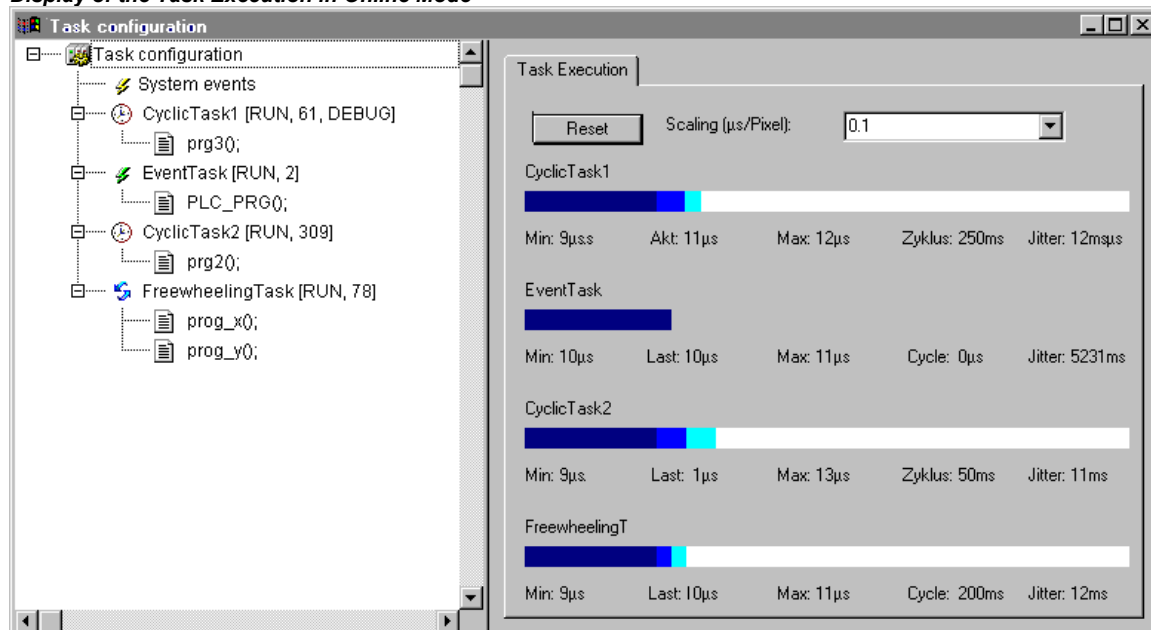
In online mode the current status of a task will be displayed in brackets at the end of the task entry line in the configuration tree, also the number of already passed through process cycles. This update interval is the same as usual for the monitoring of PLC values. The possible states:

- Idle** has not been started since last update; especially used for event tasks
- Running** has been started at least once since last update
- Stop** stopped
- Stop on BP** stopped, because breakpoint in task is reached
- Stop on Error** Error, e.g. division by zero, page fault etc.
- Stop Watchdog** cycle time has been exceeded

The task entry will be displayed red coloured in case of status 'Stop on Error' or 'Stop Watchdog' .

Display of the time flow of the tasks:

If the entry 'Taskconfiguration' is selected in the configuration tree, the utilization of the tasks will be displayed in bar charts in the right part of the window:

Display of the Task Execution in Online Mode

For each task a bar chart is displayed. The length of the bar represents the length of a cycle period. Below the bar as well as by appropriate marks on the bar the following measurement values are illustrated:

- Min:** minimum measured runtime in µs
- Akt:** last measured runtime in µs
- Max:** maximum measured runtime in µs
- Cycle:** total length of a cycle in µs
- Jitter:** maximum measured jitter (time between when the task was started and when the operating system indicates that it is running) in µs

The button **Reset** can be used to set back the values of Min., Max. and Jitter to 0.

The scaling of the chart (microseconds per Pixel) can be adjusted by the aid of a selection list at **Scaling [μ s/Pixel]**.

Which task is being processed?

For the execution, the following rules apply:

That task is executed, whose condition has been met; i.e., if its specified time has expired, or after its condition (event) variable exhibits a rising edge.

If several tasks have a valid requirement, then the task with the highest priority will be executed.

If several tasks have valid conditions and equivalent priorities, then the task that has had the longest waiting time will be executed first.

The processing of the program calls will be done according to their order (top down) in the task editor.

Depending on the target system PLC_PRG might get processed in any case as a free-wheeling task without being inserted in the task configuration tree.

Additional online functions in the context menu resp. in the 'Extras' menu:

'Extras' 'Set Debug Task'

With this command a debugging task can be set in Online mode in the Task Configuration. The text [DEBUG] will appear after the set task.

The debugging capabilities apply, then, only to this task. In other words, the program only stops at a breakpoint if the program is gone through by the set task.

'Extras' 'Enable / disable task'

With this command the task which is currently marked in the task configuration can be disabled or re-enabled. A disabled task will not be regarded during processing of the program. In the configuration tree it is indicated by a greyed entry.

'Extras' 'Callstack'

This command is available in the Extras menu in the Task Configuration. If the program is stopped at a breakpoint during debugging, it can be used to show the callstack of the corresponding POU. For this purpose the debug task must be selected in the task configuration tree of. The window 'Callstack of task <task name>' will open. There you get the name of the POU and the breakpoint position (e.g. "prog_x (2)" for line 2 of POU prog_x) . Below the complete call stack is shown in backward order. If you press button 'Go To', the focus will jump to that position in the POU which is currently marked in the callstack.

6.8 Watch and Receipt Manager

6.8.1 Overview

With the help of the Watch and Receipt Manager you can view the values of selected variables. The Watch and Receipt Manager also makes it possible to preset the variables with definite values and transfer them as a group to the PLC ('Write Receipt'). In the same way, current PLC values can be read into and stored in the Watch and Receipt Manager ('Read Receipt'). These functions are helpful, for example, for setting and entering of control parameters.

All watch lists created ('Insert' 'New Watch List') are indicated in the left column of the Watch and Receipt Manager. These lists can be selected with a mouse click or an arrow key. In the right area of the Watch and Receipt Manager the variables applicable at any given time are indicated.

In order to work with the Watch and Receipt Manager, open the object for the **Watch and Receipt Manager** in the **Resources** register card in the Object Organizer.

6.8.2 Watch and Receipt Manager in the Offline Mode

In *Offline Mode*, you can create several watch lists in the Watch and Receipt Manager using the 'Insert' 'New Watch List'.

For inputting the variables to be watched, you can call up a list of all variables with the Input Assistant, or you can enter the variables with the keyboard, according to the following notation:

```
<POUName>.<Variable Name>
```

With global variables, the POU Name is left out. You begin with a point. The variable name can, once again, contain multiple levels. Addresses can be entered directly.

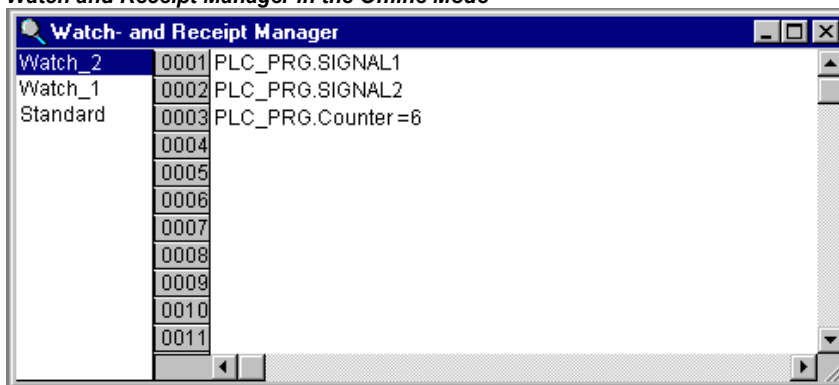
Example of a multiple-level variable:

```
PLC_PRG.Instance1.Instance2.Structure.Componentname
```

Example of a global variable:

```
.global1.component1
```

Watch and Receipt Manager in the Offline Mode



The variables in the watch list can be preset with constant values. That means that in Online mode you can use the 'Extras' 'Write Receipt' command to write these values into the variables. To do to do must use := to assign the constant value of the variable:

Example:

```
PLC_PRG.TIMER:=50
```

In the example shown in the picture above, the PLC_PRG.COUNTER variable is preset with the value 6.

Regard for variables of type **array** or **structure**: You must enter the particular elements explicitly in order to be able to preset them. Example: You have defined a structure with components a, b,c and you have declared a structure variable struvar in PLC_PRG. For pre-allocating a,b,c with values, they must be entered in the watchlist as follows:

```
PLC_PRG.struvar.a:=<value>
PLC_PRG.struvar.b:=<value>
PLC_PRG.struvar.c:=<value>
```

The presetting for the elements of an array must be done correspondingly. Example for an array variable of type ARRAY[0..6]:

```
PLC_PRG.arr_var[0]:=<value>
PLC_PRG.arr_var[1]:=<value>
...
```

If a function block fb contains the variables x,y and an instance variable fb_inst of type fb is declared in PLC_PRG, x and y can be pre-set like follows:

```
PLC_PRG.fb_inst.x:=<Wert>
PLC_PRG.fb_inst.y:=<Wert>
```

'Insert' 'New Watch List'

With this command in offline mode a new watch list can be inserted into the Watch and Receipt Manager. Enter the desired name for the watch list in the dialog box that appears.

'Extras' 'Rename Watch List'

With this command you can change the name of a watch list in the Watch and Receipt Manager.

In the dialog box that appears, enter the new name of the watch list.

'Extras' 'Save Watch List'

With this command you can save a watch list. The dialog box for saving a file is opened. The file name is preset with the name of the watch list and is given the extension "*.wtc".

The saved watch list can be loaded again with 'Extras' 'Load Watch List'.

'Extras' 'Load Watch List'

With this command you can reload a saved watch list. The dialog box is opened for opening a file. Select the desired file with the "*.wtc" extension. In the dialog box that appears, you can give the watch list a new name. The file name is preset without an extension.

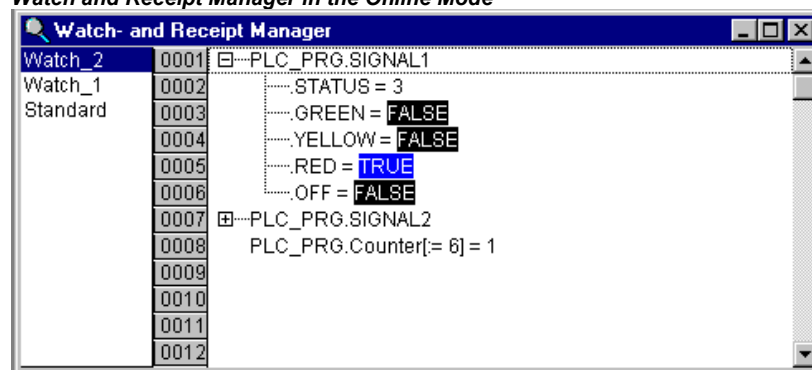
With 'Extras' 'Save Watch List', you can save a watch list.

6.8.3 Watch and Receipt Manager in the Online Mode

In Online mode, the values of the entered variables are indicated.

Structured values (arrays, structures, or instances of function blocks) are marked by a plus sign in front of the identifier. By clicking the plus sign with the mouse or by pressing <Enter>, the variable is opened up or closed. If a function block variable is marked in the watch list, the associated context menu is expanded to include the two menu items 'Zoom' and 'Open instance'.

In order to input new variables, you can turn off the display by using the 'Extras' 'Active Monitoring' command. After the variables have been entered, you can use the same command again to activate the display of the values.

Watch and Receipt Manager in the Online Mode

In the Offline Mode you can preset variables with constant values (through inputting := <value> after the variable). In the Online Mode, these values can now be written into the variables, using the 'Extras' 'Write Receipt' command.

Referring to array or structure variables please see the description in chapter 6.8.2.

With the 'Extras' 'Read Receipt' command you can replace the presetting of the variable with the present value of the variable.

Note: Only those values the watch list are loaded which was selected in the Watch and Receipt Manager!

'Extra' 'Monitoring Active'

With this command at the Watch and Receipt Manager in the Online mode the display is turned on or off. If the display is active, a check (✓) will appear in front of the menu item.

In order to enter new variables or to preset a value (see Offline Mode), the display must be turned off through the command. After the variables have been entered, you can use the same command again to activate the display of the values.

'Extras' 'Write Receipt'

With this command in the *Online Mode* of the Watch and Receipt Manager you can write the preset values (see Offline Mode) into the variables.

Note: Only those values of the watch list are loaded which was selected in the Watch and Receipt Manager!

'Extras' 'Read Receipt'

With the command, in the *Online Mode* of the Watch and Receipt Manager, you can replace the presetting of the variables (see Offline Mode) with the present value of the variables.

Example:

```
PLC_PRG.Counter [:= <present value>] = <present value>
```

Note: Only the values of that watch list are loaded which was selected in the Watch and Receipt Manager!

Force values

In the Watch and Receipt Manager you can also **'Force values'** and **'Write values'**. If you click on the respective variable value, then a dialog box opens, in which you can enter the new value of the variable. Changed variables appear in red in the Watch and Receipt Manager.


6.9 The Sampling Trace

6.9.1 Overview and Configuration

Sample tracing will be available as an object in the CoDeSys resources, if it is activated in the **target settings** (category 'General'). It can be used to trace the progression of values for variables is traced over a certain time. These values are written in a ring buffer (**trace buffer**). If the memory is full, then the "oldest" values from the start of the memory will be overwritten. As a maximum, 20 variables can be traced at the same time. A maximum of 500 values can be traced per variable.

Since the size of the trace buffer in the PLC has a fixed value, in the event of very many or very wide variables (DWORD), fewer than 500 values can be traced.

Example: if 10 WORD variables are traced and if the memory in the PLC is 5000 bytes long, then, for every variable, 250 values can be traced.

In order to be able to perform a trace, open the object for a  **Sampling Trace** in the **Resources** register card in the Object Organizer. Create resp. load an appropriate trace configuration and define the variables to be traced. (See 'Extras' 'Trace Configuration' and 'Selection of the Variables to be Displayed').

After you have created the configuration and have started the trace in the PLC ('Start Trace'), then the values of the variables will be traced. With 'Read Trace', the final traced values will be read out and displayed graphically as curves.

Please regard: If a task configuration is used for controlling the program, the trace functionality refers to the debug task.

A Trace (variable values and configuration) can be saved and reloaded in project format (*.trc) or in XML format (*.mon). Just the configuration can be stored and reloaded via a *.tcf-file.

Various traces can be available in a project for getting displayed. They are listed in a selection list ('Trace') in the upper right corner of the trace window. You can select one of those to be the currently used trace configuration.

'Extras' 'Trace Configuration'

With this command you will be given the dialog box for entering the variables to be traced, as well as diverse trace parameters for the Sampling Trace. The dialog can also be opened by a double click in the grey area of the dialog Sampling Trace.

Dialog Box for Trace Configuration

First define a name for the trace configuration (**Trace Name**). This name will be added to the selection list 'Trace' in the upper right corner of the Trace window, as soon as you have confirmed and closed the configuration dialog with OK. Optionally enter a **Comment**.

The list of the **Variables** to be traced is initially empty. In order to append a variable the variable must be entered in the field under the list. Following this, you can use the **Insert** button or the <Enter> to append the variable to the list. You can also use the Input Assistant (**Help Manager**). The use of enumeration variables is possible.

A variable is deleted from the list when you select the variable and then press the **Delete** button.

A Boolean or analogue variable (also an enumeration variables) can be entered into the field **Trigger variable**. The input assistance can be used here. The trigger variable describes the termination condition of the trace.

In **Trigger Level** you enter the level of an analogue trigger variable at which the trigger event occurs. You also can use an ENUM constant here. When **Trigger edge positive** is selected the trigger event occurs after an ascending edge of the Boolean trigger variable or when an analogue variable has passed through the trigger level from below to above. **negative** causes triggering after a descending edge or when an analogue variable went from above to below. **both** causes triggering for both descending and ascending edges or by a positive or negative pass, whereas **none** does not initiate a triggering event at all.

Trigger Position is used to set the percentage of the measured value which will be recorded before the trigger event occurs. If, for example, you enter 25 here then 25 % of the measured values are shown before the triggering event and 75% afterwards and then the trace is terminated.

The field **Sample Rate** is used set the time period between two recordings in milliseconds resp., if supported by the target system, in microseconds. The default value "0" means one scanning procedure per cycle.

Select the mode for recalling the recorded values (**Recording**): With **Single** the Number of the defined samples are displayed one time. With **Continuous** the reading of the recording of the defined number of measured values is initiated anew each time. If, for example, you enter the number '35' the first display contains the first measured values 1 to 35 and the recording of the next 35 measured values (36-70) will then be automatically read, etc.. **Manual** selection is used to read the trace recordings specifically with 'Extras' 'Read trace'.

The recall mode functions independently of whether a trigger variable is set or not. If no trigger variable is set the trace buffer will be filled with the defined number of measured values and the buffer contents will be read and displayed on recall.

The button **Save** is used to store the trace configuration which has been created in a file. The standard window "File save as" is opened for this purpose.

Stored trace configurations can be retrieved using the button **Load**. The standard window "File open" is opened for this purpose.

Note:	Please note that Save and Load in the configuration dialog only relates to the configuration, not to the values of a trace recording (in contrast to the menu commands 'Extras' 'Save trace values' 'Save values' and '..'Load values').
--------------	--

If the field **Trigger Variable** is empty, the trace recording will run endlessly and can be stopped by by 'Extras' 'Stop Trace'.

Selection of the Variables to be displayed

The combo box es to the right, next to the window for displaying curves trace variables defined in the trace configuration. If a variable is selected from the list, then after the trace buffer has been read the variable will be displayed in the corresponding color (Var 0 green, etc.). Variables can also be selected if curves are already displayed.

A maximum of up to eight variables can be observed simultaneously in the trace window.

6.9.2 Generating a Trace Sampling

'Extra' 'Start Trace'

Symbol: 

With this command the trace configuration is transferred to the PLC and the trace sampling is started in the PLC.

'Extra' 'Read Trace'

Symbol: 

With this command the present trace buffer is read from the PLC, and the values of the selected variables are displayed.


Use the commands of this menu to save or reload traces (configuration + trace values) in files resp. from files, to load a trace from the controller to the project or to set a certain trace as that which should be used in the project.

Note: Regard the alternative way of storing and reloading traces by using the commands of menu 'Extras' 'Save trace values' (Project format, *.trc-Datei, ASCII) !

'Extra' 'Auto Read'

With this command the present trace buffer is read automatically from the PLC, and the values are continuously displayed. If the trace buffer is automatically read, then a check (ü) is located before the menu item.

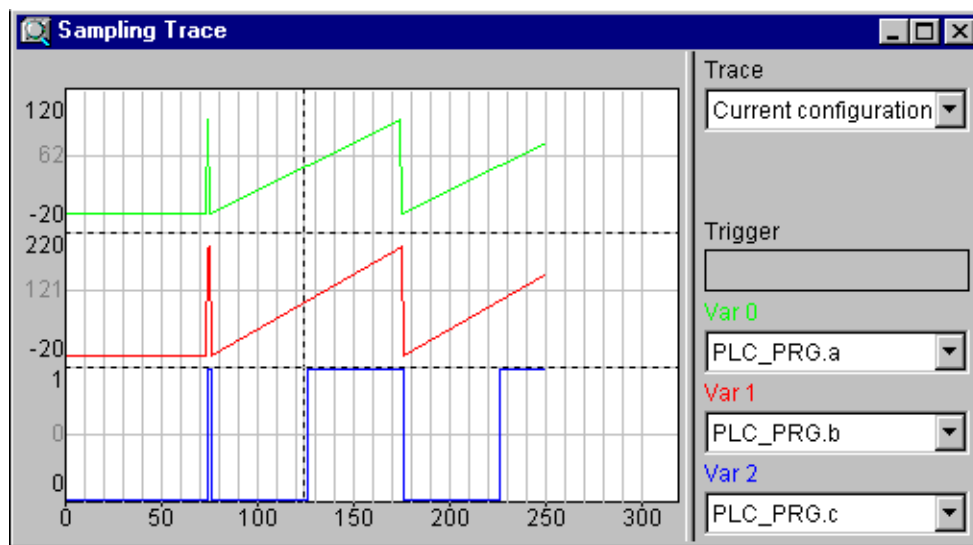
'Extra' 'Stop Trace'

Symbol: 

This command stops the Sampling Trace in the PLC.

6.9.3 Looking at the Sampling Trace

Sampling Trace of Different Variables



If a trace buffer is loaded ('Extras' 'Start Trace'), then the values of all variables to be displayed can be read out ('Extras' 'Read Trace' or 'Extras' 'Auto Read') and will be displayed accordingly. If no scan frequency has been set, then the X axis will be inscribed with the continuous number of the traced value. The trace buffer will be deleted as soon as the trace sampling is stopped ('Extras' 'Stop Trace').

The status indicator of the trace window (first line) indicates whether the trace buffer is full and when the trace is completed.

If a value for the scan frequency was specified, then the x axis will specify the time of the traced value. The time is assigned to the "oldest" traced value. In the example, e.g., the values for the last 25 seconds are indicated.

The Y axis is inscribed with values in the appropriate data type. The scaling is laid out in a way that allows the lowest and the highest value to fit in the viewing area. In the example, Var 0 has taken on the lowest value of 6, and the highest value of 100: hence the setting of the scale at the left edge.

If the trigger requirement is met, then a vertical dotted line is displayed at the interface between the values before and after the appearance of the trigger requirement.

'Extras' 'Cursor Mode'

The easiest way to set a cursor in the monitoring area is to click there with the left mouse button. A cursor appears and can be moved by the mouse. At the top of the monitoring window the current x-position of the cursor is displayed. In the fields next to 'Var 0', 'Var 1', ..., 'Var n' the value of the respective variable is shown.

Another way is the command 'Extras' 'Cursor mode'. With this command two vertical lines will appear in the Sampling Trace. First they are laying one on the other. One of the lines can be moved to the right or to the left by the arrow keys. By pressing <Ctrl>+<left> or <Ctrl>+<right> the speed of the movement can be increased by factor 10.

If additionally the <Shift> key is pressed, the second line can be moved, showing the difference to the first one.

'Extras' 'Multi Channel'

With this command you can alternate between single-channel and multi-channel display of the Sampling Trace. In the event of a multi-channel display, there is a check (✓) in front of the menu item.

The multi-channel display has been preset. Here the display window is divided into as many as eight display curves. For each curve the maximum and the minimum value are displayed at the edge.

In a single-channel display, all curves are displayed with the same scaling factor and are superimposed. This can be useful when displaying curve abnormalities.

'Extras' 'Show grid'

With this command you can switch on and off the grid in the graphic window. When the grid is switched on, a check (✓) will appear next to the menu item.

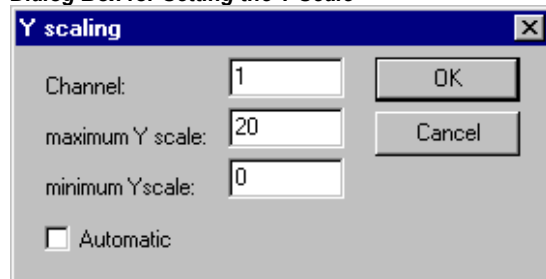
'Extras' 'Y Scaling'

With this command you can change the preset Y scaling of a curve in the trace display. By doubleclicking on a curve you will also be given the dialog box 'Y-scaling'.

As long as option **Automatic** is activated, the default scaling will be used, which depends on the type of the used variable. In case of enumeration variables the enumeration values will be displayed at the scale. In order to change the scaling, deactivate option 'Automatic' and enter the number of the respective curve (**Channel**) and the new maximum (**maximum y scale**) and the new minimum value (**minimum y scale**) on the y axis.

The channel and the former value are preset.

Dialog Box for Setting the Y Scale



'Extras' 'Stretch'

Symbol: 

With this command you can stretch (zoom) the values of the Sampling Trace that are shown. The beginning position is set with the horizontal picture adjustment bar. With repeated stretches that follow one-after-another, the trace section displayed in the window will increasingly shrink in size.

This command is the counterpart to 'Extras' 'Compress'.

'Extras' 'Compress'

Symbol: 

With this command the values shown for the Sampling Trace are compressed; i.e., after this command you can view the progression of the trace variables within a larger time frame. A multiple execution of the command is possible.

This command is the counterpart to 'Extras' 'Stretch'.

6.9.4 'Extras' 'Save trace values'

Use the commands of this menu to save traces (configuration + values) to files resp. to reload them from files to the project. Besides that a trace can be saved in a file in ASCII-format.

Note: Regard the alternative way of storing and reloading traces by using the commands of menu 'Extras' 'External Trace Configurations' (XML format, *.mon-file) !

'Save Values'

With this command you can save a Sampling Trace (values + configuration data). The dialog box for saving a file is opened. The file name receives the extension **"*.trc"**.

Be aware, that here you save the traced values as well as the trace configuration, whereas **Save trace** in the configuration dialog only concerns the configuration data.

The saved Sampling Trace can be loaded again with 'Extras' 'Load Trace'.

'Load Values'

With this command a saved Sampling Trace (traced values + configuration data) can be reloaded. The dialog box for opening a file is opened. Select the desired file with the **"*.trc"** extension.

With 'Extras' 'Save Values' you can save a Sampling Trace.

'Trace in ASCII-File'

With this command you can save a Sampling Trace in an ASCII-file. The dialog box for saving a file is opened. The file name receives the extension **"*.txt"**. The values are deposited in the file according to the following scheme:

```
CODESYS Trace
D:\CODESYS\PROJECTS\TRAFFICSIGNAL.PRO
Cycle PLC_PRG.COUNTER PLC_PRG.LIGHT1
0 2 1
1 2 1
2 2 1
.....
```

If no frequency scan was set in the trace configuration, then the cycle is located in the first column; that means one value per cycle has been recorded at any given time. In the other respects, the entry here is for the point in time in ms at which the values of the variables have been saved since the Sampling Trace has been run.

In the subsequent columns, the corresponding values of the trace variables are saved. At any given time the values are separated from one another by a blank space.

The appertaining variable names are displayed next to one another in the third line, according to the sequence (PLC_PRG.COUNTER, PLC_PRG.LIGHT1).

6.9.5 'Extras' 'External Trace Configurations'

'Save to file'

With this command a trace (configuration + values) can be saved in a file in XML format. For this purpose the standard dialog for saving a file opens. Automatically the file extension *.mon will be used.

A *.mon-file can be reloaded to a project with command 'Load from file'.

'Load from file'

With this command a trace (configuration + values), which is available in a file in XML format (*.mon), can be loaded into the project. The dialog for opening a file will open and you can browse for files with extension *.mon. The loaded trace will be displayed and added to the selection list in field 'Trace in the configuration dialog. If you want to set it as currently used project trace configuration, use command 'Set as project configuration'.

A *.mon-file can be created by using command 'Save to file'.

Note: Regard the alternative way of saving a trace by using the commands of menu 'Extras' 'Save trace values'.

'Save to target'

With this command in online mode a trace configuration, which is available in a file in XML format (*.mon), can be loaded into the controller (target).

The standard dialog for opening a file will be available, where per default files with extension *.mon will be displayed. Regard in this concern the possibility to save trace configurations in *.mon-files (XML format) 'Save to file'.

See also: 'Load from target'

'Load from target'

With this command the trace (configuration + values) which is currently used on the controller (target) can be loaded to the CoDeSys project. It will be displayed in the trace window and can be set as active project trace configuration.

'Set as project configuration'

With this command the trace configuration which is currently selected in the list of available traces (field 'Trace' in the trace window) can be set as active configuration within the project. The selection list besides the currently used (top position) offers all traces which have been loaded to the project by command 'Load from file' from *.mon-files (e.g. for the purpose of viewing).

6.10 Workspace

This object in the 'Resources' tab provides an image of the currently set project options (see chapter 4.2, Project Options). If you open it, you get the 'Options' dialog with the know categories.

6.11 Parameter Manager

The Parameter Manager is a target specific component of the CoDeSys programming system and must be activated in the target settings. (see chapter 6.11.1).

The Parameter Manager can be used to make variables of a CoDeSys IEC-program, constant parameters or specific system parameters accessible to all CoDeSys compatible systems in a network for the purpose of data exchange, typically via field bus. For this purpose in the editor you can create parameter lists and load down to and up from the runtime system.

Please regard: Parameter lists also can be created resp. filled with entries via pragmas which are included in variable declarations (see chapter 5.2.3).

What are Parameters ?:

In this context parameters are:

process variables of the CoDeSys IEC project

process independent parameters

specific system parameters, predefined by the target system

function block instances or structure variables, arrays

Each parameter is identified by a certain set of **attributes** like e.g. 'default value', 'access rights', and especially by an unique **access key** ('Index', 'SubIndex', 'Name'), which can be addressed for reading or writing data from/to the parameter list. This data exchange can be done via communication services and it is not necessary to know any addresses of variables or to provide any extra functions. So the use of the Parameter Manager functionality it is an alternative to using Network Variables.

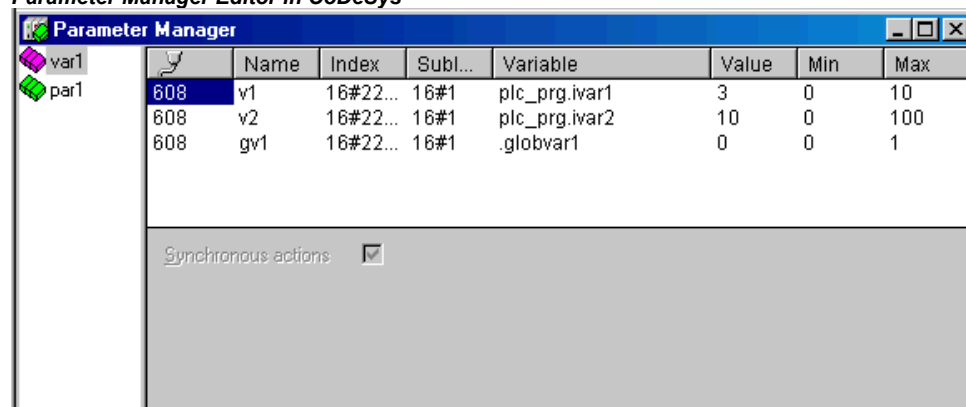
What are Parameter Lists?:

Parameter lists are used for organizing the parameters and can be saved with the project and loaded to the local target system which is controlled by the corresponding IEC-program. For each type of parameters there is a corresponding type of parameter list.

Each parameter entry is represented by a line in the parameter list. Each **column** of the list is representing one of the parameter attributes. In addition to a certain set of standard attributes also manufacturer specific attributes might be used for the description of a parameter in the Parameter Manager.

It depends on the definitions in a **target specific description file** which attributes (columns) will be visible and editable in the Parameter Manager and in which way they will be arranged in a parameter list. If the description file is missing, the complete standard set of attributes will be displayed, each showing the default value.

Parameter Manager Editor in CoDeSys



Besides lists for project variables and project constants the Parameter Manager also can handle lists for system parameters. Those are parameters which are given by the target system. Further on you

can create lists for function block instances or structure variables which base on user-defined **templates** also created in the Parameter Manager.

Due to the fact that the data are stored independently of the IEC-program, a parameter list for example can be used for saving 'recipes', which are preserved even if the program is replaced by another version. Further on a running PLC can be "fed" with different recipes without the need of a re-download the program.

Note: It is depending on the target system, whether the parameter manager will be regarded at the creation of a **boot project**.

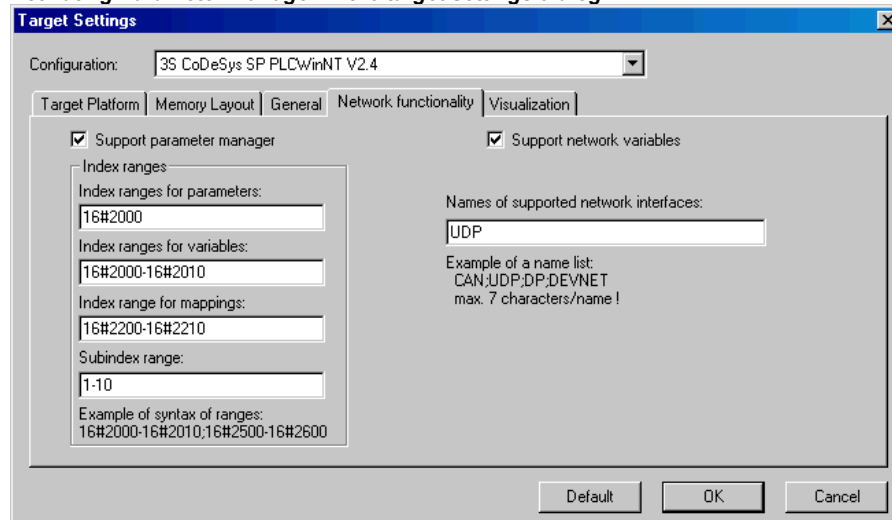
6.11.1 Overview, Activating

The Parameter Manager must be activated in the **Target Settings**, Category **Network functionality** (see Chapter 6.12).

Also in the target settings dialog the index and subindex ranges for the entries in parameter lists of type parameters and variables, and – if supported by the target – mappings (for CAN Device PDOs) must be defined.

It depends on the target system, whether these options are visible resp. editable for the user.

Activating Parameter Manager in the target settings dialog



6.11.2 The Parameter Manager Editor, Overview

In the Resources tab choose the object 'Parameter-Manager'. An editor window will open, where you can create, edit and store parameter lists and in online mode also can load them to the target system and monitor the current parameter values.

Note: In order to have available the Parameter Manager functionality in a CoDeSys project, the option 'Support Parameter Manager' in the Target Settings must be activated and appropriate index ranges must be defined there !

The editor window is bipartite. The left part is used for navigation, it shows a list of all parameter lists currently loaded to the Parameter Manager. The right part contains a table editor, the columns titled with the names of the attributes.

In the **navigation window** you insert, delete, rearrange or rename parameter lists of different types (Variables, Constant Parameters, Template, Instance, System Parameters).

In the **table editor** you fill the lists with parameter entries. Each list type will show a special selection of attribute columns which can be edited or just are visible. Either this is defined by a target specific description file or the default settings will be taken.

You can jump between navigation window and table editor by pressing **<F6>**.

Parameter Manager Editor in CoDeSys

	Name	Index	Subl...	Variable	Value	Min	Max
608	v1	16#22...	16#1	plc_prg.ivar1	3	0	10
608	v2	16#22...	16#1	plc_prg.ivar2	10	0	100
608	gv1	16#22...	16#1	.globvar1	0	0	1

Synchronous actions

In online mode you can load the lists, you have created before, to the currently connected target system. You also can use the Parameter Manager functions to access them there for the purpose of data exchange with other systems (write values, upload). Further on in the Parameter Manager window you can monitor the current values of the parameters. If currently no online connection is established, the parameter lists just can be created locally and saved with the project.

6.11.3 Parameter List Types and Attributes

The Parameter Manager can handle the following parameter list types:

- **Variables:** The entries in parameter lists of this type represent process variables of the project.
- **Parameters:** The entries in parameter lists of this type represent parameters whose values are not attached by the process.
- **System parameters:** The entries in parameter lists of this type represent data which are not attached by the process and which are determined by the target system. System Parameter lists cannot be deleted or renamed.
- **Template:** A template does not contain parameter entries which can be directly accessed for the purpose of data exchange. In fact the entries provide a "basic attribute configuration" for the components of a function block or a structure. Thus a template can be used in parameters lists of type 'Instance'.
- **Instance:** The entries in parameter lists of this type represent parameter entries for variables which are of type of a function block or structure, that means which are instances or structure variables. For an easy entering of the parameters a template is used, which has also been created in the Parameter Manager before.
- **Mappings:** This list type is only available, if it is supported by the target system. The entries represent process variables which are intended to be used in the PDO mapping of a CAN-Device. So mapping lists basically are variables list, but they are working on a separate index/subindex range. This range must be defined in the target settings, category Network functionality! In this case a CAN-Device, which is configured in the PLC Configuration, **only** will use the entries of list type 'Mapping', while otherwise all entries of variables or instances lists will be available in the PDO mapping dialog.

Each list type will be displayed in the Parameter Manager Editor according to the attributes defined by a description file in XML format. If such a file is missing, default settings will be used.

Instances and Templates

An "Instance" parameter list ...

... handles parameter entries, which represent a **function block**, a **structure** variable or an **array**. Instance lists for a function block or a structure are each based on a **template** which is also to be defined in the Parameter Manager for the respective function block resp. structure. Instance lists for arrays cannot use a template made in the Parameter Manager, but directly refer to the array which is used in the project.

A "Template" parameter list ...

... does not contain parameters which are directly accessed for the purpose of data exchange. In fact it defines index and subindex offsets and certain attributes for parameter entries which represent the components of a function block or a structure. The template then can be used in a 'Instance' parameter list (see above), thus providing an easy way to create parameter entries for project variables which are instances of a function block or a structure.

Creating a Template parameter list:

In the edit field next to **Base POU** enter the name of the function block or structure for which a parameter template should be created. Using the input assistant you can browse the available POUs of the project. Press **Apply** to enter the components of the chosen POU in the parameter list editor. Now edit the attribute fields and close the list to make it available for use in an 'Instance' list.

The command **Insert missing entries** in the context menu or in the 'Extras' menu will cause an update of the entries according to the current version of the Base POU. This might be necessary after having deleted some lines or after having changed the Base-POU.

If option **Synchronous actions** is activated, all read-/write accesses on other POUs defined for any list entries, will be executed by the target system synchronously with the call of the respective entry.

For creating Instance parameter lists for **arrays** it is not necessary to create an template in the Parameter Manager. The template ARRAY will be available implicitly.

Creating an Instance parameter list:

Edit a **Template** from the selection list below the table. This list offers all templates currently available for function blocks or structures in the Parameter Manager plus the option ARRAY, which you select, if you want to refer directly to an array used in your project. Press **Apply** to insert the predefined components to the parameter list table.

In the edit field **Base variable** enter exactly that project variable (must be of type of the function block or the structure or the array which is described by the chosen template), for the components of which you want to create parameter entries.

Enter a **Base index** and **Base subindex** for the instance. The indices and subindices of the particular components then will be calculated automatically by adding the index resp. subindex values which are defined in the template for each component (in case of arrays the base will be 0). They will be filled automatically to the respective attribute fields. Example: If you enter a base index "3" for a component, for which an index offset "3000" is defined in the template, the component will be set to index 3003.

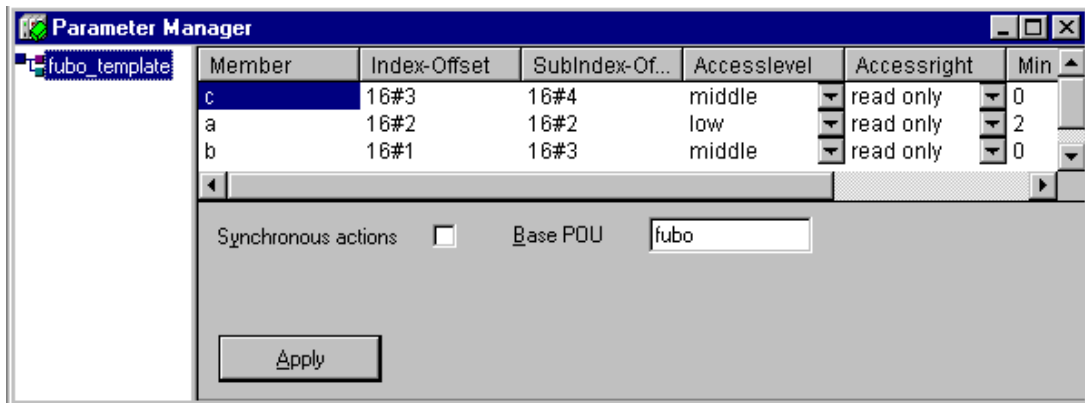
For option **Synchronous actions** please see above: Creating a Template parameter list.

The command **Insert missing entries** in the context menu or in the 'Extras' menu will cause an update of the entries according to the current version of the used template. That might be useful after having deleted entries or after the template has been modified.

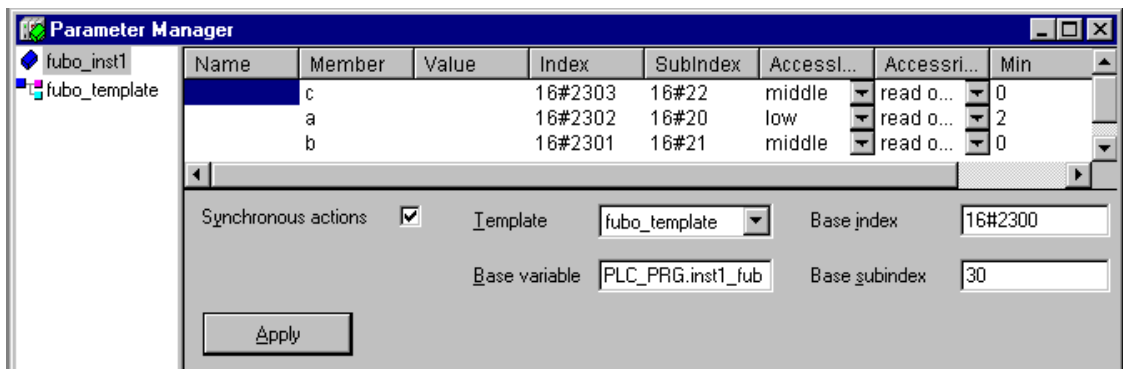
Example:

Create a function block fubo with input- or output variables: a,b,c. In PLC_PRG define the following FB-instances: inst1_fubo:fubo; inst2_fubo:fubo. Compile the project.

Now open the Parameter Manager in order to create parameter lists for the variables inst1_fubo.a, inst1_fubo.b, inst1_fubo.c and inst2_fubo.a, inst2_fubo.b, inst2_fubo. First insert a parameter list which is of type 'Template' and name it "fubo_template". Define the Base-POU: "fubo". Press Apply and define some attributes for the components a,b,c: te. Inter alia enter the Index offsets: for a: 16#1, for b: 16#2, for c: 16#3. Also the SubIndex offsets, e.g. a: 16#2, b: 16#3, c: 16#4.



Now insert a new parameter list which is of type 'Instance'. Choose template "fubo_template". Insert the Base variable "inst1_fubo". Define a Base index of e.g. 16#2300 and a Base subindex of 30 (you must regard the ranges defined in the target settings in tab Networkfunctionality !). Now press Apply to get displayed the indices which are calculated for the components a, b, c by the addition of base offsets and template defined offsets: Indices: 16#2301, 16#2302, 16#2303; SubIndices:16#23, 16#33, 16#43.



Basing on this automatically created entries now you can continue to edit the parameter list.

6.11.4 Managing parameter lists

Insert list

Shortcut: Ins

To insert a new parameter list in the Parameter Manager use the command 'Insert list...', resp. 'Insert new list' in the 'Insert' or context menu. The commands are available when the focus is in the empty navigation window resp. on an already existing entry in the navigation tree. The dialog 'Insert list' will open.

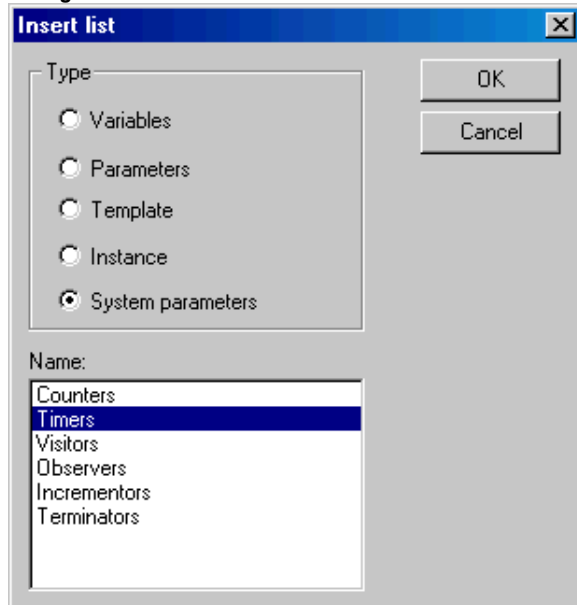
Insert a Name for the new parameter list (must be unique within the list type) and choose one of the following list types:

Variables	Entries for process variables
Parameters	Entries for data, whose values remain unattached by the process
Template	Template of attribute setting for the components of a function block or a structure
Instance	Entries for variables of type of a function block or a structure, basing on the corresponding template (see above)
Mappings	Entries for process variables, intended for being used in the PDO mapping of a CAN Device. This type is only available, if supported by the target system !

System parameters

Entries for parameters whose values are not attached by the process and which are defined by the target system

Dialog 'Insert list'



After confirming the settings and closing the dialog with **OK** the new parameter list will appear as an entry in the navigation window, the list type indicated by the icon. In the table editor the appropriate attributes will be displayed as column titles. Selection and order of the columns are defined by a target specific description file, otherwise the default settings are used. Now you can edit the table, by entering a line for each desired parameter (see Chapter 6.11.4, Editing a parameter list).

Rename List

The parameter list, which is currently marked in the navigation window, can be renamed by using the command 'Rename list' which is available in the 'Extras' menu or in the context menu. An edit field will open, which you also get when doing a double-click with the mouse on the list name.

Cut / Copy /Paste list

Shortcut: <Ctrl> + <X>, <Ctrl> + <C>, <Ctrl> + <V>,

The command 'Cut' (Edit menu) resp. 'Cut list' (context menu) moves the currently marked list from the navigation window to a temporary buffer, so that you can reinsert it by the 'Paste' command at any other position in the navigation tree. Before re-inserting mark that list, above which you want to insert.

The command 'Copy' resp. 'Copy list' also uses the temporary buffer, but the original navigation tree entry will be kept, and a copy will be added by 'Paste'.

Delete list

Shortcut:

The list currently selected in the navigation window will be removed by the command 'Delete' ('Edit' Menu) resp. 'Delete list' ('Extras' or context menu).

Please regard: In online mode this command will delete the corresponding list in the runtime system.

6.11.5 Editing parameter lists

Which columns (attributes) are displayed

The currently marked parameter list (navigation window) will be displayed in the table window as defined by a target specific description file resp. according to the default settings.

This means that the attributes' values of each particular parameter will be displayed in a separate **line** according to the list-type-specific order and selection of columns.

You can **fade out** and **fade in** columns by deactivating/activating them in the context menu when the cursor is pointing to any field of the list column title bar.

For modifying the column move the dividers between the column title fields or use one of the commands available in the context menu when the cursor is placed on a column title field: Command **Standard column width** will set a standard width for all columns which makes them all visible in the window. **Maximize width** will change the width of the currently focussed column so that each entry will be fully displayed.

Commands for editing a parameter list entry

The following commands for editing a parameter list are available in the **context menu** resp. in the menus '**Insert**' or '**Extras**':

Inserting /Deleting lines:

Insert line resp. New line	A new entry (line) will be inserted before that one where the cursor is currently placed.
Line after resp. New line after Shortcut:<Ctrl><Enter>	A new entry (line) will be inserted after that one where the cursor is currently placed. .
Delete line Shortcut: <Shift>+	The line, where the cursor is currently placed, will be deleted.
Cut, copy, paste line	These commands can be used to move (cut/paste) or to copy (copy/paste) the selected line.

Editing attribute values:

If a new line for a parameter entry is inserted, the attribute fields will be automatically filled with target specific default values. See chapter 6.11.3, 'Parameter List Types and Attributes' for the possible attributes. To enter or edit an attribute value, click on the corresponding field. An edit field will be opened, if the attribute is editable. The input assistant (<F2>) will be available in fields where a component of the CoDeSys project should be entered.

Press <Enter> to close the entry.

Use the arrow keys to jump to another field.

Press to delete the entry of the currently edited field.

In order to toggle the input format between 'decimal' and hexadecimal' use the command '**Format Dec/Hex**' in the '**Extras**' menu.

Press <F6> in order to set the focus to the navigation window (and back).

Options:

Below the table in the editor window there can be activated the following options (availability depending on list type):

Download with program: At a program download the list will be downloaded automatically to the controller.

Synchronous actions: All read-/write accesses on other POU's defined for any list.

entries, will be executed by the target system synchronously with the call of the respective entry.

Sorting parameter lists

The sequence of entries within a parameter list can be sorted concerning an attribute (column) in ascending or descending order of the attribute values. This works in offline and in online mode.

Perform a mouse-click on the field which contains the column title of the desired attribute. Thus the table lines will be re-sorted and in the column title field of the attribute an arrow symbol will be displayed, showing the current sorting (pointing upwards = ascending sort sequence, pointing downwards = descending sort sequence).

6.11.6 Parameter Manager in Online Mode

List transfer between Editor and Controlling Unit

If supported by the target, in online mode the parameter lists, which have been created in the editor, can be **downloaded** to resp. **uploaded** from the runtime system. Further on you can **write single parameter values** to the runtime system. The maximum size of lists of type 'Variable' and 'Parameters' also is defined by the target system.

Please regard: At login automatically a download of all parameter lists will be done for which the option 'Load with project' is activated !

The current value of each parameter is **monitored** in an additional column which is displayed in the parameter manager in online mode :



It depends on the target, whether Index and Subindex or RefID and Offset are used for monitoring the values.

The following commands are available in the 'Extras' menu for handling the list transfer between editor and controller:

- | | |
|-----------------------------|--|
| Delete list | The list currently marked in the navigation window will be deleted from the PLC runtime system. |
| Write list | This command will open the dialog 'Copy objects' where you can select from the available lists those you want to download to the runtime system. The download will be done as soon as you confirm with OK. It depends on the target whether for enumerations only the numeric or additionally the symbolic values will be transferred. |
| Read list | All lists of type 'Parameters' will be read from the runtime system and loaded into the Parameter Manager. The "upload" of lists of type 'Variables' will be done only if explicitly supported by the target. |
| Write values | All values defined in column 'Value' will be written to the parameter list in the runtime system. To write single values, perform a double-click on the respective field in the column to get the dialog 'Write value', as known from the function 'Online' 'Write values'. |
| Write default values | The values defined in column 'Default' will be written to the parameter list in the runtime system. |
| Take over values | The current values will be read from the runtime system and be uploaded to column 'Value'. |

The command **Format Dec/Hex** is also available to toggle the input format between 'decimal' and hexadecimal' .

Parameter lists in boot project

It depends on the target system, whether parameter lists will be regarded when a boot project is created.

6.11.7 Export / Import of parameter lists

'Extras' 'Export'

The command 'Export' of the 'Extras' menu can be used to export the lists of the Parameter Manager to a XML-file. This file for example might be imported to another project by using the import function in the CoDeSys Parameter Manager. The standard dialog for saving a file will be opened, the file extension *.prm will be preset. All lists available in the Parameter Manager will be written to the export file.

The content of the Parameter Manager also can be exported using the general project export function ('Project' Export').

'Extras' 'Import'

The command 'Import' of the 'Extras' menu can be used to import a XML-file which describes parameter lists. This file for example might be created by using the export function in the CoDeSys Parameter Manager.

If the import file contains a list, the name of which is already used in the Parameter Manager, a dialog will open asking the user whether the existing list should be overwritten.

6.12 Target Settings

The "Target Settings" is an object of the 'Resources'. Here you define, which target shall be used for the project and how it will be configured. If a new project is started with '**Project** **New**', a dialog will open where the target, that means a predefined configuration for a target, has to be set.

The list of available targets depends on which Target Support Packages (TSP) are installed on the computer. These describe platform specific basic configurations and also define, to which extent the configuration can be modified by the user in the **CoDeSys** Target settings dialogs.

Please regard: If no TSP is available, only the setting '**None**' will be offered in the target system selection box. This will switch to simulation mode automatically and no configuration settings will be possible.

Target-Support-Package

A Target Support Package (TSP) must be installed before starting by the aid of the installation program **InstallTarget** which might be part of the **CoDeSys**-Setup.

A Target Support Package (**TSP**) contains all the files and configuration information necessary to control a standard platform with a program created in **CoDeSys**. What has to be configured: codegenerator, memory layout, PLC functionality, I/O modules. In addition libraries, gateway drivers, ini-files for error messaging and PLC browser have to be linked. The central component of a TSP is one or more **Target files**. A Target file directs to the files which are in addition necessary to configure the target. It is possible that several target files share these additional files.

The default extension for a Target file is *.trg, the format is binary. Additive definitions are attached to the entries in the target file which determine whether the user can see and edit the settings in the **CoDeSys** dialogs.

During installation of a TSP the target file for each target is put to a separate directory and the path is registered. The associated files are copied to the computer according to the information of a **Info file** *.tnf. The name of the target directory is the same as the targets name. It is recommended to store the target specific files in a directory which is named with the manufacturers name.

The files which get installed with a TSP are read when **CoDeSys** is started. The target settings which are done in the **CoDeSys** dialogs will be saved with the project.

Please note: If you use a new target file or if you have changed the existing one, **CoDeSys** has to be restarted to read the updated version.

Dialog Target Settings

The dialog **Target Settings** will open automatically, when a new project is created. It also can be opened by selecting the menu item 'Target Settings' in the register 'Resources' in the Object Organizer.

Choose one of the target configurations offered at **Configuration**.

If no Target Support Package has been installed, only '**None**' can be selected, which means working in simulation mode. If you choose one of the installed configurations it depends on the entries in the target files, which possibilities are left to customize this configuration in the **CoDeSys** dialogs. If you choose a target configuration, for which there exists no valid licence on the computer, **CoDeSys** asks you to choose another target.

If a configuration is selected, which is provided with the entry "HideSettings" in the corresponding target file, you only can see the name of the configuration. Otherwise there are five dialogs available to modify the given configuration:

1. Target Platform
2. Memory Layout
3. General
4. Networkfunctionality
5. Visualization

Attention !: Please be aware, that each modification of the predefined target configuration can cause severe changes in performance and behaviour of the target !

Press <Default> if you want to reset the target settings to the standard configuration given by the target file.

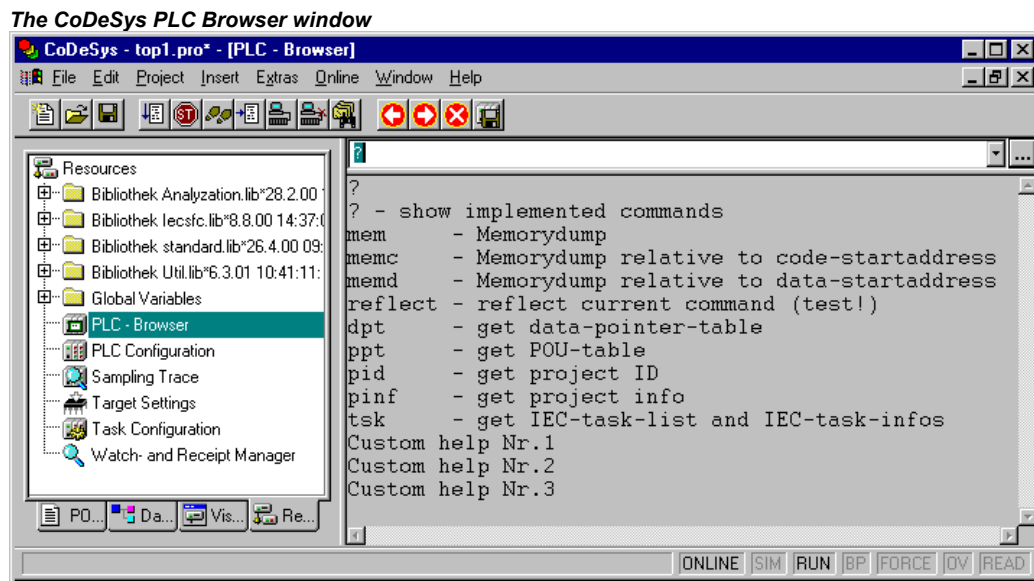
6.13 The PLC-Browser

The PLC-Browser is a text-based control monitor (terminal). Commands for the request of specific information from the controller are entered in an entry line and sent as string to the controller. The returned response string is displayed in a result window of the browser. This functionality serves diagnostic- and debugging purposes.

The commands available for the set target-system are made up of the **CoDeSys** standard set plus a possible extension set of the controller manufacturer. They are managed in an ini file and implemented accordingly in the runtime system.

6.13.1 General remarks concerning PLC-Browser operation

Select the entry PLC-Browser in the Resources tab-control. It will be available there if it is activated in the current target settings (category networkfunctionality).



The browser consists of a command entry line and a result/display window.

In a selection box the input line displays a list of all the commands entered since the start of the project (input history). They are available for re-selection until the project is closed. Only commands, which differ from those already existing, are added to the list.

The entered command is sent to the controller with <Enter>. If there is no Online connection, the command is displayed in the result window in the same way as it is sent to the controller, otherwise the response from the controller is shown there. If a new command is sent to the controller, the content of the result window is deleted.

Commands can be entered in the form of command strings, the use of macros is possible as well:

6.13.2 Command entry in the PLC-Browser

Basically the PLC-Browser makes available the **3S standard commands** hard-coded in the run-time system. It is concerned with functions for direct memory manipulation, for the output of project- and status functions as well as for run-time monitoring. They are described in the browser's **ini-file**, which is an integral part of the Target Support Package. These standard commands can be further supplemented by specialized ones, e.g. self-diagnostic functions or other status messages of the control application. The expansion of the command list must be carried out both in the customer interface in the run-time system as well as through additional entries in the Browser ini-file.

When opening the project the **command list** available in the PLC-Browser is generated based on the entries in the Browser ini-file. It can be accessed as input help using the ... key in the dialog „Insert standard command" or using <F2>. Also the command **'Insert' 'Standard commands'** can be used to get the command list. A command can be typed in manually to the command line or it can be selected from the list by a double-click on the appropriate entry.

The general command syntax is:

```
<KEYWORD><LEER><KEYWORD-DEPENDEND PARAMETERS>
```

The keyword is the **command**. With which **parameters** it can be expanded is described in the respective tooltip in the entry help window.

The command, which has been sent, is repeated in the output data window, the controller's response appears below it.

Example: Request for the project Id from the controller with the command "pid"

Entry in command line:

```
Pid
```

Output in result window:

```
pid
Project-ID: 16#0025CFDA
```

A **help text** can be supplied for each standard command with ?<BLANK><KEYWORD>. This is similarly defined in the ini-file.

The following commands are firmly integrated in the run-time system and contained in the ini-file with the corresponding entries for entry help, tooltips and help:

Command	Description
?	The run-time system supplies a list of the available commands. The list is independent of the status of the description files of the target system.
mem	Hexdump of a memory area Syntax 1: mem <start address> <end address> Addresses can be entered decimal, hexadecimal (Prefix 16#) or as a macro.
memc	Hexdump relative to the start address of the code in the controller; like mem, the data are added to the code area start address.
memd	Hexdump relative to the data base address in the controller; like mem, the data are added to the data area
reflect	Reflect current command line, for test purposes.
dpt	Read and display data-pointer table.
ppt	Read and display POU table.
pid	Read and display project Id.
pinf	Read and display project info (see 'Project' 'Project Info').
tsk	Show list of IEC-tasks incl. task infos defined in the project.
startprg	Start PLC program ('Online' 'Start').
stopprg	Stop PLC program ('Online' 'Stop').
resetprg	Reset PLC program. Only not-retentive data get initialized. ('Online' 'Reset').
resetprgcold	Reset PLC program cold. Retentive data also get initialized. ('Online' 'Reset (cold)').
resetprgorg	Reset PLC program original. The current application program as well as all data (incl. retentive and persistent) are deleted. ('Online' 'Reset (origin)').
reload	Reload boot project.
getprgprop	Read and display program properties (Name, title, version author, date).
getprgstat	Read and display program status (e.g. "run", "stop", last error, flags).
filedir	File command "dir". List of files in the PLC directory.
filecopy	Copy file [from] [to]. Example: "filecopy filename.txt filename2.txt".
filerename	Rename files on PLC [old] [new]. Example: filerename oldname.txt newname.txt".
filedelete	Delete file on PLC; Example: "filedelete file.xml".
saveretain	Save retain variables. The name of the save file will be displayed afterwards.
restoreretain	Load retain variables. The name of the save file, from which the variables values are restored, will be displayed.

Command	Description
setpwd	Set password on controller; Syntax: setpwd <password> [level], e.g. "setpwd abcde 0". <level> can be "0" (default) just valid concerning logins from the programming system, or "1" valid for all applications.
delpwd	Delete password on PLC.

Please regard:

The first word of the command sequence entered is interpreted as **keyword** (<KEYWORD>).

If the first word of the command entry is **not recognized** by the controller, the response 'Keyword not found' will appear in the result window.

If a keyword is preceded by a „?**SPACE**“ (e.g. „? mem“), the ini-file will be searched for the existence of a help section to this keyword. If one is available, nothing is sent to the controller, but only the help text is displayed in the output data window.

6.13.3 Use of macros during the command entry in PLC-Browser

If a command associated with a macro is entered in the command line, this is expanded before it is sent to the controller. Then the response in the result window appears in a similarly expanded form.

The entry syntax is: <KEYWORD><macro>

<KEYWORD> is the command.

Macros are:

%P<NAME> If NAME is a POU-name, the expression is expanded to <POU-Index>, otherwise there is no alteration

%V<NAME> If NAME is a variable name, the expression is expanded to #<INDEX>:<OFFSET>, otherwise there is no alteration (this notation #<INDEX>:<OFFSET> is interpreted by the controller as a memory address)

%T<NAME> If NAME is a variable name, the expression is expanded to <VARIABLENTYP>, otherwise there is no alteration.

%S<NAME> If NAME is a variable name, the expression is expanded to <SIZEOF(VAR)>, otherwise there is no alteration.

The % character is ignored if the escape symbol \ (Backslash) is placed in front. The escape symbol as such is only transmitted if written \\.

Example:

Entry in command line: (memory dump of the variable .testit ?)

```
mem %V.testit
```


Output in result window:


```
mem #4:52
03BAAA24 00 00 00 00 CD CD CD CD ....íííí
```

6.13.4 Further PLC-Browser options

In the 'Extras' menu or in the PLC-Browser's toolbar there are the following commands for handling the command entry or history list:

With **History forward**  and **History backward**  you can scroll backwards and forwards through the query results already carried out. The history recording is continued until you leave the project.

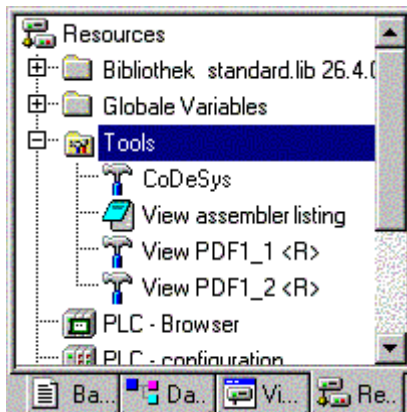
With **Cancel command**  you can break off a query which has been initiated.

With **Save history list**  you can save the query results carried out up until that point in an external text file. The dialogue 'Save file as' will appear, in which you can enter a file name with the extension „.bhl" (Browser History List). The command **Print last command** opens the standard dialogue to print. The current query plus the output data in the message window can be printed.

6.14 Tools

The object 'Tools' will be available in the 'Resources' tab if the functionality is enabled for the currently set target system. It shows all available shortcuts (connections) to executable files of external tools, which can be activated by a double-click in order to call these external programs from within CoDeSys. It is defined by the target file which and how many shortcuts are allowed. Depending on this definition the user can add or delete new shortcuts in the 'Tools' folder.

For example the Tools folder in the Object Organizer might look like this:



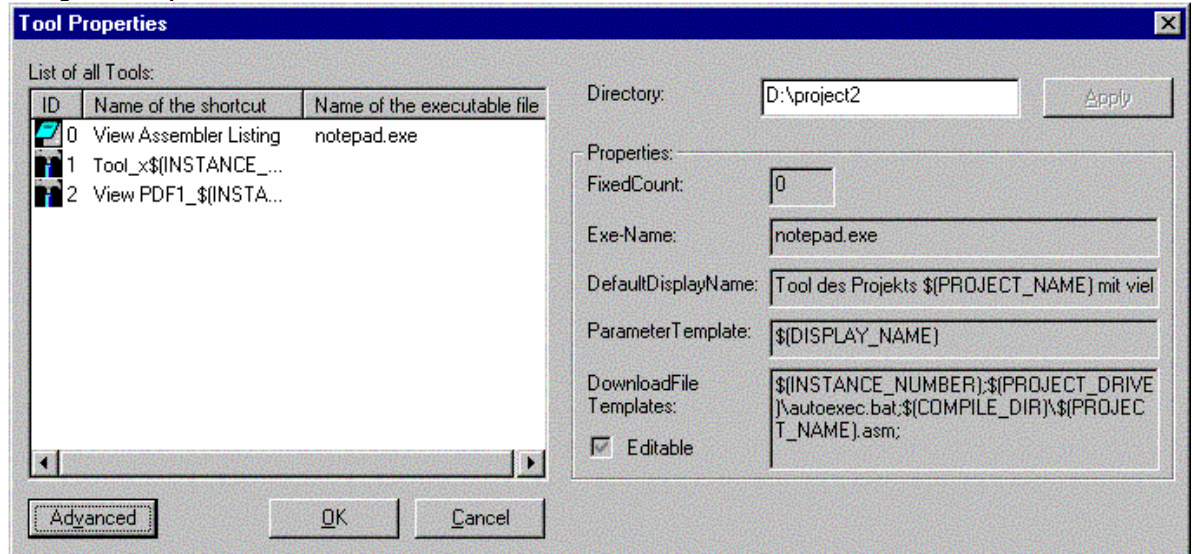
In this example four tools-shortcuts are installed. One serves for starting another CoDeSys programming system, one for opening the assembler listing in a text editor and two further shortcuts are available to open PDF-files. Shortcuts marked with a "<R>" cannot be modified in CoDeSys. The shortcuts may contain the connection to an editor, e.g. notepad.exe, or to a certain PDF-file, so that a double-click on the entry would open a notepad window showing the assembler listing respectively would open the Acrobat Reader showing the PDF-file.

Additionally you can define certain files which should be downloaded to the PLC as soon as the shortcut is activated.

6.14.1 Properties of available Tool Shortcuts (Object Properties)

By a mouse-click on the plus sign at entry 'Tools' in the Resources tab of the Organizer a list of the available shortcuts will open. If you are just starting to set up a new project, you will just see those which are defined in the target file as fix entries. But if the Tools folder already had been modified you might find another shortcuts, added by a user in CoDeSys.

Dialog 'Tool Properties'



You can view the global tool properties (valid for all shortcuts listed in 'Tools') as well as the properties of single shortcuts.

1. Tool Properties:

If 'Tools' is marked in the Resources tree, you will find the command 'Object Properties' in the context menu or in the menu 'Project' 'Object', which will open the dialog 'Tool Properties'.

There you get a table listing all tool shortcuts which might be used for the currently set target. The following parameters are shown: The **Id** (unique identification number) of the tool, the **Name of the shortcut** which is used to reference the shortcut in the Object Organizer, and the **Name of the executable file** of the tool. The button **Advanced** expands the dialog resp. closes the extension of the dialog:

The expanded dialog shows the global properties of the shortcut as defined in the target file. Further on an edit field is available where a (working) **Directory** can be defined which should be used for actions of the executable file. The path will be saved without closing the dialog as soon as you press the **Apply** button.

Properties of the Tool:

- FixedCount** Number of shortcuts of the tool, which are inserted unalterably and automatically in the Tools folder. Only if "0" is entered here, the user will be able to create as many shortcuts as desired.
Please regard: For shortcuts which are defined as "fix" ones by the target file, the number of possible usage in the Tools folder is predetermined and the properties cannot be modified by the CoDeSys user (cognizable by a "<R>" in the Object Organizer).
- Exe-Name:** File name or full path of the executable file of the tool. Here you also can enter a registry path pointing to an exe-file: "[registry path].<registry entry in this path pointing to an exe-file>" If there is no entry, the file extension of the file, which is given in "Parameter Template", automatically will cause via Windows the start of the exe file of the according tool.
Examples: "C:\programme\notepad.exe", "345.pdf"
- DefaultDisplayName:** Name which is used to represent the tools in the Object Organizer. Possibly the template \$(INSTANCE NUMBER) is used (see below 'Parameter Template').
- Parameter Template:** Templates for determining the file which should be opened by the tool. The following templates can be used, connected by the appropriate special characters:
\$(PROJECT_NAME) Name of the currently opened project
(File name without extension *.pro").
\$(PROJECT_PATH) Path of the directory where the project file is

(without indication of the drive).

\$(PROJECT_DRIVE) Drive where the currently opened project is.

\$(COMPILE_DIR) Compile directory of the project (including indication of the drive)

\$(TOOL_EXE_NAME) Name of the exe-file of the tool.

\$(DISPLAY_NAME) Name of the current shortcut, as used in the 'Tools' folder.

\$(INSTANCE_NUMBER) Number of the shortcut
(Instance number, running number, starting with "1")

\$(CODESYS_EXE_DIR) Path of the directory where the Codesys exe-file is
(including indication of the drive).

The conversion of a template you will see in the dialog for the Shortcut Properties
(see below)

Example:

"\$(PROJECT_NAME)_\$(INSTANCE_NUMBER).cfg"

The cfg-file with the name <name of current CoDeSys project>_<shortcut number>.cfg will be opened in the tool.

DownloadFile Templates:

Files, file paths resp. templates for file which will be copied to the PLC during download.. If option **Editable** is activated, the list of these files will be editable in the properties dialog of the shortcut. If a file name is entered without path, the file will be searched in the directory where the codesys-exe-file is.

Example:

"a.up;\$(PROJECT_NAME).zaw;\$(INSTANCE_NUMBER).upp"

the files a.up, <current CoDeSys Projekt>.pro and <shortcut number>.upp will be copied to the PLC during the next download

2. Shortcut Properties:

Mark a shortcut entry in the 'Tools' tree in the Object Organizer and select the command 'Object Properties' in the context menu or in the 'Project' 'Object' menu. The dialog 'Shortcut Properties' will open, containing the following items:

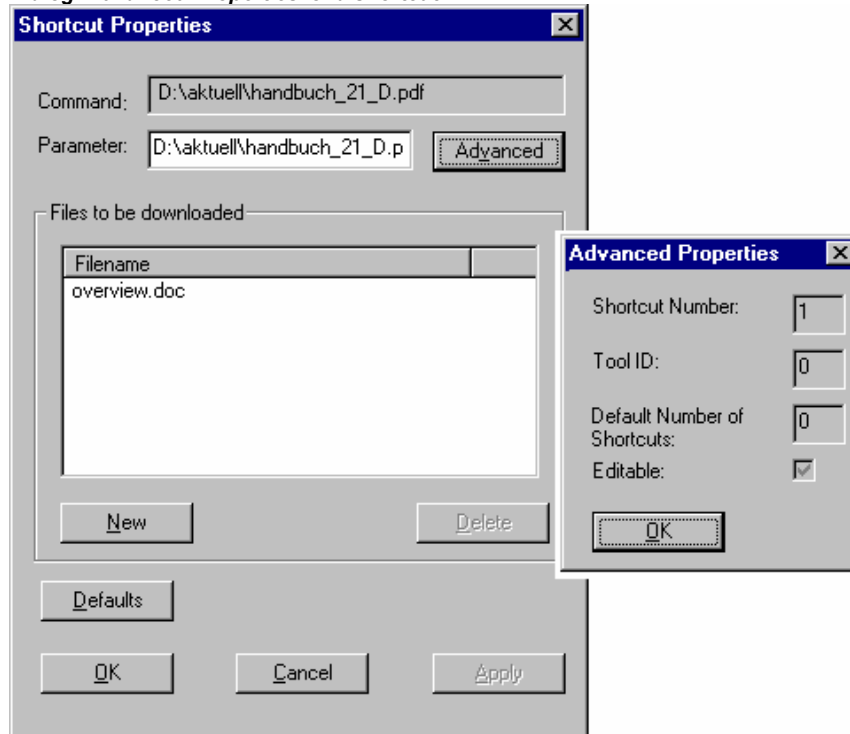
Command	Calling the tool; paths of the exe-file and of the file which is named in 'Parameter' (predefined by the 'Parameter Template', see above) e.g.: C:\programs\notepad.exe D:\listings\textfile.txt
Parameter	Path of the file which should be called by the tool. This is defined in the target file and can be edited here, if the option 'Editable' (see below) has been activated.
Files to be downloaded	Primarily you find here the Filenames which are defined by the target file and which are also described in the Tool Properties (DownloadFileTemplate, see above). If option 'Editable' is activated in the extended dialog (see below) then you can modify the list. For this purpose press button New to open the dialog ' Filename ', where you can enter another file resp. a file path. If you enter a file without path, then it will be searched in that directory, where the codesys-exe-file is. Button Delete will remove the currently marked list entry.

Button **Standard** resets the entries of the dialog to the default values defined by the target file.

Button **Apply** saves the done settings without closing the properties dialog.

Button **Advanced** expands the dialog so that it will look as follows:

Dialog 'Advanced Properties' of a shortcut



- Shortcut Number:** Running number, starting with 1. New shortcuts of the current tool will each get the next higher number. If a shortcut will be removed later, the numbers of the remaining shortcuts will stay unchanged. The shortcut number can be inserted in other definitions by using the template $\$(INSTANCE_NUMBER)$ (e.g. see above, 'Parameter Template').
- Tool ID:** Unique identification number of the tool; defined in the target file.
- Default Number of Shortcuts:** Number of shortcuts (instances) for a tool. Corresponds to the "FixedCount" defined in the target file. See above, Tool Properties.
- Editable:** If this option is activated, it will be possible to edit the field 'Parameter' resp. the list of files which should be downloaded.

Button **OK** applies the done settings and closes the Properties dialog.

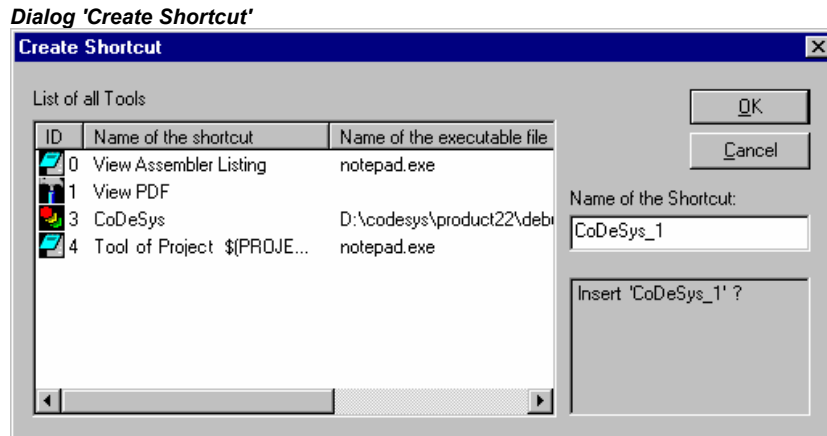
6.14.2 Managing Tool Shortcuts

Creating new Tool Shortcuts

Select the entry 'Tools' or a shortcut entry in the Resources tree of the Object Organizer and select command 'Add Object' in the context menu or in the 'Project' 'Object' menu to open the dialog '**Create Shortcut**', see below.

The table lists all tools for which new shortcuts (connections) can be created. According to the definitions in the target file the following parameters are displayed: **ID** of the tool, default **Name of the shortcut** and the **Name of the executable file**.

In order to create a(nother) shortcut for one of the offered tools, select this tool by a mouse-click in the 'ID' column. Hereupon you can modify the default name of the shortcut in the edit field **Name of the shortcut** and confirm with OK. This will only work if you enter a name which is not yet used.



OK closes the dialog and the new shortcut will be inserted in the Resources tree, represented by the shortcut name and a shortcut number which is 1 higher than the highest one used for a instance of this tool up to now.

In the area below the name field appropriate hints concerning the user inputs will be displayed.

Deleting Tool Shortcuts

Deleting a shortcut can be done via the command Delete in the context menu (right mouse button) or in the 'Project' 'Object' menu. The command is only available, if there is no fix number of shortcuts defined for the currently marked shortcut. The shortcut numbers of the remaining shortcuts will not change if you delete a shortcut.

Executing Tool Shortcuts

A shortcut will be executed on a double-click on the entry in the Resources tree or by the command 'Open Object' in the 'Project' 'Object' menu resp. in the context menu (right mouse button).

If the execution of the file, which is defined in the shortcut properties (Parameter), fails, then an appropriate error message will appear. If a parameter file will not be found, the exe-file of the tool will be executed and a dialog will open, asking you whether the file should be created.

If the exe-file of the tool is not found in the defined path or if no path has been defined, then the standard dialog for selecting a file will be opened and the user will be asked to enter the path of the exe-file. This path will be saved when the dialog is closed by OK and thus will be available for the tool also in other CoDeSys projects.

Saving Tool Shortcuts

When the CoDeSys project is saved, the status and settings of the 'Tools' folder in the Resources tree will also be saved.

Please note: If you save a project by 'Save as' with a new name, then you must consider the following if you use the template `$(PROJECT_NAME)` in the definition of the parameter file and of the files which are to be downloaded:
If you had added shortcuts for a tool (FixedCount=0) in the old project, then in the new project the file names have to be renamed manually corresponding to the new project name. In contrast for a tool which is defined with a fix number of shortcuts, the template always will be replaced automatically by the current project name !

6.14.3 Frequently asked questions on Tools

Why do I get no entry 'Tools' in the 'Resources' ?

Only if it is defined in the target file of the currently set target system, the 'Tools' functionality will be available.

For which tools already shortcuts are available, which shortcuts can I add in the CoDeSys project ?

Open the folder 'Tools' in the 'Resources' tree of the Object Organizer by a double-click on the plus sign. You will see which tools already are connected for the current project. If you have just set up a new project and not yet worked on the Tools list, then just those entries will be displayed, which are predefined unalterably by the definitions in the target file. Otherwise you might see an already project specifically modified tools list. In order to check, whether the list is extendable by new entries, select the command 'Add Object'. You will get a dialog offering all tools for which additional shortcuts can be created.

Which global properties do the available tools have ?

Mark the entry 'Tools' in the Object Organizer and choose the command 'Object Properties' from the context menu (right mouse button). Expand the appearing dialog by pressing the 'Advanced' button. Now you will see a list of the available tools and the corresponding parameters. Select one of the tools by a mouse click on the ID-Symbol in order to – for example - get displayed the allowed number of shortcuts for the tool in the field 'FixedCount', or to get displayed which files will be downloaded to the PLC if the shortcut is activated. The file names or paths might be shown in the form of templates, which will be interpreted for each single shortcut as described in the following paragraph:

Which individual properties have the available shortcuts ?

Mark one of the entries below 'Tools' in the Object Organizer and select the command 'Object Properties' in the context menu (right mouse button). Press button 'Advanced' to get the parameters of the chosen shortcut. Partially they will correspond to the above described global tool properties. If allowed by the definition in the target file you can edit these parameters here.

How can I create a shortcut for a tool ?

Mark the entry 'Tools' in the Object Organizer and choose the command 'Add Object' from the context menu (right mouse button). You will see a list of available tools, but only of those for which the maximum number of shortcuts (FixedCount) is not yet reached. Choose a tool and press OK. The tool will now be inserted in the Tools folder in the Object Organizer. If you want to insert it once more, then you must enter a different tool name first, which means to mark the new entry as another instance of the same tool. For example you could name the instances of the tool Toolxy "Toolxy_1", "Toolxy_2" etc.

How can I modify the parameters of a tool ?

In order to modify the parameters of a shortcut (instance of a tool connection), mark the shortcut in the Object Organizer and choose command 'Object Properties' from the context menu. It depends on the pre-definition of the tool in the target file, whether the parameters can be edited in the properties dialog. (See in the expanded dialog whether the option 'Editable' is activated. Button 'Standard' resets all edited values to the defaults.

How can I execute a tool shortcut ?

Perform a double-click on the shortcut entry in the Object Organizer or select the command 'Open Object' in the context menu resp. in the 'Project' menu when the entry is marked in the Object Organizer.

Tools

7 ENI

7.1.1 What is ENI

The ENI ('Engineering Interface') allows to connect the CoDeSys programming system to an **external data base**. There the data which are needed during creation of an automation project can be stored. The usage of an external data base guarantees the consistency of the data, which then can be shared by several users, projects and programs. Also it extends the CoDeSys functionality by making possible the following items:

Revision control for CoDeSys projects and associated resources (shared objects): If a object has been checked out from the data base, modified and checked in again, then in the data base a new version of the object will be created, but the older versions will be kept also and can be called again on demand. For each object and for a whole project the version history will be logged. Two versions can be checked for differences.

Multi-User Operation: The latest version of a sample of objects, e.g. of POU's of a project, can be made accessible for a group of users. Then the objects which are currently checked out by one of the users will be marked as "in the works" and not editable by the other users. Thus several users can work in parallel on the same project without risking to overwrite versions mutually.

Access by external tools: Besides the CoDeSys programming system also other tools, which have an ENI too, can access the common data base. These might be e.g. external visualizations, ECAD systems etc., which need the data created in CoDeSys or which also produce data which are needed by other programs.

The ENI is composed of a **client** and a **server** part. So it is possible to hold the data base on a remote computer, which is required for multi-user operation. The CoDeSys programming system is a client of the independent **ENI Server Process** as well as another application, which needs access to the data base (Please see the separate documentation on the *ENI Server*).

Currently the ENI supports the data base systems 'Visual SourceSafe 6.0', 'MKS Source Integrity', 'PVCS Version Manager' V7.5 and higher and a local file system. Objects can be stored there in different 'folders' (data base categories with different access properties). The objects can be checked out for editing and thereby will get locked for other users. The latest version of an object can be called from the data base. Further on in parallel you can store any objects just locally in the project as usual for projects without source control.

7.1.2 Preconditions for Working with an ENI project data base

Please regard: For a guide concerning installation and usage of the *ENI Server* provided by 3S – Smart Software Solutions GmbH please see the separate server documentation resp. online help. There you will also find a quick start guide.
Also consider the possibility of using the *ENI Explorer* which allows to perform data base actions independently from the currently used data base system.

If you want to use the ENI in the CoDeSys programming system in order to manage the project objects in an external data base, the following preconditions must be fulfilled:

the communication between CoDeSys and the ENI Server requires **TCP/IP**, because the *ENI Server* uses the HTTP-Protocol.

an *ENI Server (ENI Server Suite)* must be installed and started locally or on a remote computer. A license is required to run it with one of the standard database drivers which has been installed with the server. Just the driver for a local file system can be used with a non-licensed ENI Server version.

in the *ENI Server* service control tool (**ENI Control**) the connection to the desired data base must be configured correctly (Data base). You will automatically be asked to do this during installation, but you can modify the settings later in *ENI Control*.

a **project data base** for which an ENI-supported driver is available, must be installed. It is reasonable to do this on the same computer, where the *ENI Server* is running. Alternatively a local file system can be used, for which a driver will also be provided by default.

in the **data base administration** possibly the user (Client) as well as the ENI Server must be registered as valid users with access rights. Anyway this is required for the 'Visual SourceSafe', if you use another data base system, please see the corresponding documentation for information on the user configuration.

for the current CoDeSys project the **ENI interface must be activated** (to be done in the CoDeSys dialog 'Project' 'Options' 'Project data base'). (It is possible to switch in a user definition in ENI, e.g. for the purpose of defining more detailed access rights as it is possible in the data base administration. But in general it is sufficient if the user who wants to log in to the data base via ENI, is registered in the data base.

for the current CoDeSys project the **connection to the data base** must be configured correctly; this is to be done in the CoDeSys dialogs 'Project' 'Options' 'Project source control'.

in the current project the user must **log in to the ENI Server** with user name and password; this is to be done in the Login dialog, which can be opened explicitly by the command 'Project' 'Data Base Link' 'Login' resp. which will be opened automatically in case you try to access the data base without having logged in before.

7.1.3 Working with the ENI project data base in CoDeSys

The data base commands (Get Latest Version, Check Out, Check In, Version History, Label Version etc.) which are used for managing the project objects in the ENI project data base, will be available in the current CoDeSys project as soon as the connection to the data base has been activated and configured correctly. See for this the Preconditions for Working with an ENI project data base . The commands then are disposable in the submenu 'Data Base Link' of the context menu or of the 'Project' menu and refer to the object which is currently marked in the Object Organizer.

The current assignment of an object to a data base category is shown in the Object Properties and can be modified there.

The properties of the data base categories (communication parameters, access rights, check in/check out behaviour) can be modified in the option dialogs of the project data base ('Project' 'Options' 'Project Source Control').

7.1.4 Object categories concerning the project data base

There are four categories of objects of a CoDeSys project concerning the project source control:

The ENI distinguishes three categories ("ENI object categories") of objects which are managed in the project data base: Project objects, Shared objects, Compile files.

The category 'Local', to which an object will be assigned if it should not be stored in the data base. That means that it will be handled as it is known for projects without any source control.

Thus in the programming system a CoDeSys object can be assigned to one of the categories 'Project objects', 'Shared objects' or 'Local'; the 'Compile files' do not yet exist as objects within the project. Assigning an object to one of the categories is done automatically when the object is created, because this is defined in the project options dialog 'Project source control', or explicitly via command 'Project' 'Data Base Link' 'Define' or 'Multiple Define', but it can be modified anytime in the 'Object Properties' dialog.

Each ENI object category will be configured separately in the settings for the 'Project source control' which are part of the project options ('Project' 'Options'). That means that each category gets defined own parameters for the communication with the data base (directory, port, access right, user access data etc.) and concerning the behaviour at calling the latest version, checking out and checking in. These settings then will be valid for all objects belonging to the category. As a result you will have to log in to each data base category separately; this will be done via the 'Login' dialog.

It is advisable to create a separate folder for each object category in the data base, but it is also possible to store all objects in the same folder. (The 'category' is a property of an object, not of a folder.)

See in the following the three ENI object categories:

- Project Objects:** Objects which contain project specific source information, e.g. POUs which are shared in a multi-user operation. The command 'Get all latest versions' automatically will call all objects of this category from the data base to the local project; even those, which have not been there so far.
- Shared Objects:** Objects which are not project specific, e.g. POU libraries which normally are used in several projects.
Attention: The command 'Get all Latest Versions' only will copy those objects of this category from the project folder to the local project, which are already part of the project !
- Compile files:** Compile information (e.g. symbol files) which is created by CoDeSys for the current project and which may be needed by other programs too. Example: An external visualization may need not only the project variables, but also the assigned addresses. The latter will not be known until the project is compiled.

Alternatively any objects of the CoDeSys project can be excluded from the project source control and can be assigned to category 'Local', which means that they are just stored with the project as usual for projects without any source control.

Tools

8 DDE Interface

DDE Communication with CoDeSys

CoDeSys has a DDE (dynamic data exchange) interface for reading data. **CoDeSys** uses this interface to provide other applications that also use a DDE Interface with the contents of control variables and IEC addresses

If the GatewayDDEServer is used, which works with symbols, **CoDeSys** is not needed to read variables values from the PLC and to transfer them to applications with an DDE interface.

Attention: Direct addresses cannot be read via the DDE Server! For this case in CoDeSys variables with the appropriate address assignment (AT) have to be declared.

Attention: The DDE interface has been tested with Word 97 and Excel 97 on Windows NT 4.0. If the DDE communication fails caused by a mismatch of other versions or additionally installed programs on a computer, 3S – Smart Software Solutions cannot take any responsibility.

8.1 DDE interface of the CoDeSys programming system...

Activating the DDE Interface

The DDE interface becomes active as soon as the PLC (or the simulation) is logged in.

General Approach to Data

A DDE inquiry can be divided into three parts:

1. Name of the program (here: **CoDeSys**),
2. File name and
3. Variable name to be read.

Name of the program: **CoDeSys**

File name: complete project path (c:\example\example.pro).

Variable name: The name of a variable as it appears in the Watch and Receipt Manager .

Which variables can be read?

All addresses and variables are readable. Variables or addresses should be entered in the format used in the Watch and Receipt Manager

Examples:

%IX1.4.1 (* Reads the input 1.4.1*)

PLC_PRG.TEST (* Reads the variable TEST from the POU PLC_PRG*)

.GlobVar1 (* Reads the global variable GlobVar1 *)

Linking variables using WORD

In order to get the current value of the variable TEST from the POU PLC_PRG through the DDE interface in Microsoft WORD, a field (e.g., the date) must be inserted in WORD ('Insert' "Field"). Now when you click on the field with the right mouse button and select the command "Toggle Field Codes" you can change the field function for the chosen text. In our example, this would look as follows:

```
{ DDEAUTO CODESYS "C:\CODESYS\PROJECT\IFMBSP.PRO" "PLC_PRG.TEST" }
```

Click on the field with the right mouse button again, then click on "Update Field" and the desired variable content appears in the text.

Linking variables using EXCEL

The following must be entered in Microsoft EXCEL before you can assign a variable to a cell.

```
=CODESYS | 'C:\CODESYS\PROJECT\IFMBSP.PRO' ! PLC_PRG.TEST'
```

When you click on 'Edit' then "Links", the result for this link will be:

Type: CODESYS

Source file: C:\CODESYS\PROJECT\IFMBSP.PRO

Element: PLC_PRG.TEST

Accessing variables with Intouch

Link with your project a DDE Access Name <AccessName> with the application name CODESYS and the DDE topic name C:\CODESYS\PROJECT\IFMBSP.PRO.

Now you can associate DDE type variables with the access name <AccessName>. Enter the name of the variable as the Item Name (e.g., PLC_PRG.TEST).

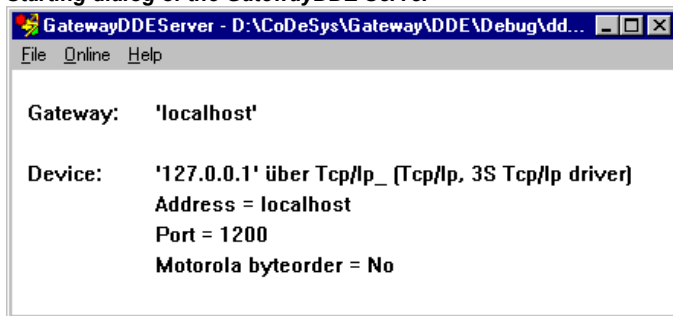
8.2 DDE communication with the GatewayDDE Server...

Handling of the GatewayDDE Server

The GatewayDDE Server can use the symbols which are created in **CoDeSys** for a project to communicate with other clients or the PLC. (see 'Project' 'Options' 'Symbolconfiguration'). It can serve the DDE interfaces of applications like e.g. Excel. This allows to transmit the variables values of the PLC to an applications, e.g. for the purpose of monitoring.

At start of the GatewayDDE Server a window opens, where the configuration of start and communication parameters can be done. A already existing configuration file can be called or the parameters can be set newly.

Starting dialog of the GatewayDDE Server

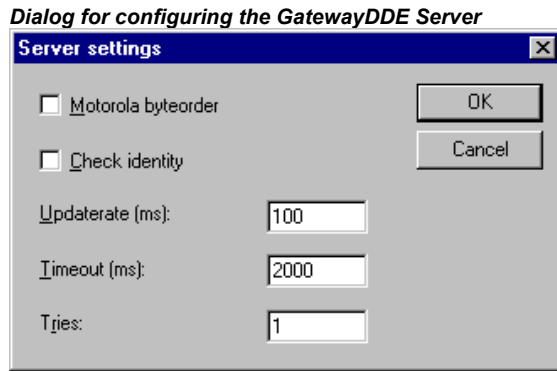


Using the command '**File**' '**Open**' you can call an already existing file which stores a set of configuration parameters. The standard dialog for selecting a file will open and available files with the extension ".cfg" will be offered. If a configuration file is selected, the configuration parameters and the defined target device are displayed

If the option '**File**' '**Autoload**' is activated, the GatewayDDE Server automatically opens with that configuration, which was active before the last terminating of the server.

If the server is started without any predefined configuration and without the setting Autoload, then in the configuration window 'Gateway:' und 'Device:' are displayed. Then you have to set up a new configuration.

The command '**File**' '**Settings**' opens the dialog '**Server settings**', in which the following parameters can be set:



Motorola byteorder	Motorola Byteorder used
Check identity	It will be checked, whether the project ID given by the symbol file is the same as that stored in the PLC.
Updaterate [ms]	Time interval for reading all symbol values from the PLC.
Timeout [ms]	Communication timeout for the used driver.
Tries	Number of retries of the communication driver to transfer a data block (not supported by all drivers !)

To set up the connection to the Gateway, the dialog '**Communication Parameters**' is opened by the command 'Online' 'Parameters'. It is the same dialog as you get in **CoDeSys** with the command 'Online' 'Communication parameters'. The settings you do here must be the same as in the corresponding **CoDeSys** Project.

The actual configuration of the GatewayDDE Server can be stored in a file by the command '**File**' '**Save**'. The standard dialog for saving a file will open, default for the extension of the file is *.cfg.

To get the gateway in active mode, log in by the command '**Online**' '**Login**'. (The gateway symbol in the status bar will get lightened then.) At login the desired connection will be built up and the available symbols can be accessed.. These must have been created before in the **CoDeSys** Project!

To log out use the command '**Online**' '**Logout**'.

General Approach to Data

A DDE inquiry can be divided into three parts:

1. Name of the program
2. File name
3. Variable name to be read

Name of the program: **GatewayDDEServer**

File name: name of the project from which variables should be read (e.g. example.pro).

Variable name: The name of a variable as it appears in the Watch and Receipt Manager .

Which variables can be read?

All variables can be read. The variables should be entered in the format used in the Watch and Receipt Manager. Regard that direct addresses cannot be read!

Examples:

PLC_PRG.TEST (* Reads the variable TEST from the POU PLC_PRG*)

.GlobVar1 (* Reads the global variable GlobVar1 *)

Linking variables using WORD

Start the GatewayDDEServer before activating the inquiry in WORD.

In order to get the current value of the variable TEST from the POU PLC_PRG through the DDE interface in Microsoft WORD, a field (e.g., the date) must be inserted in WORD ('Insert' "Field"). Now when you click on the field with the right mouse button and select the command "Toggle Field Codes" you can change the field function for the chosen text. In our example, this would look as follows:

```
{ DDEAUTO GATEWAYDDESERVER "BSP.PRO" "PLC_PRG.TEST" }
```

Click on the field with the right mouse button again, then click on "Update Field" and the desired variable content appears in the text.

Linking variables using EXCEL

Start the GatewayDDEServer before activating the inquiry in EXCEL.

The following must be entered in Microsoft EXCEL before you can assign a variable to a cell.

```
=GATEWAYDDESERVER|<Dateiname>!<Variablenname>
```

Beispiel:

```
=GATEWAYDDESERVER|'bsp.pro'!'PLC_PRG.TEST'
```

When you click on 'Edit' then "Links", the result for this link will be:

Type: CODESYS

Source file: C:\CODESYS\PROJECT\IFMBSP.PRO

Element: PLC_PRG.TEST

```
{ DDEAUTO GATEWAYDDESERVER "BSP.PRO" "PLC_PRG.TEST" }
```

Command line options for the GatewayDDEServer

If the GatewayDDE Server is started by a command line, the following options can be attached:

- | | | | |
|----|--|------------------|------------------|
| /n | The info dialog does not appear automatically at starting | | |
| /s | Display of the dialog window | /s=h | No |
| | | /s=i | minimized (icon) |
| | | /s=m | maximized |
| | | /s=n | normal |
| /c | Configuration file to be load automatically | /c=<config-file> | |
| /o | Go online with the selected configuration (autoload or defined by "/c=") | | |

Example:

Command line:

```
GATEWAYDDE /s=i /c="D:\DDE\conf_1.cfg"
```

The GatewayDDE Server will be started, the dialog window will appear as an icon, the configuration which is stored in the file conf_1.cfg will be loaded.

9 The License Management in CoDeSys

9.1 The License Manager

The 3S License Manager is available to handle the licenses for 3S modules, as well as licenses for modules for which an appropriate license information file is provided, on your computer. In CoDeSys you can create a project and provide it as a licensed library. The Licensing Manager will be installed automatically with any 3S module, which requires a license.

See also:

Separate documentation provided with the *3S Licensing Manager*

Creating a licensed library in CoDeSys, Chapter 9.1.1.

9.1.1 Creating a licensed library in CoDeSys

As is known a CoDeSys project can be saved as a library. If you want to create a licensed library you have to add the appropriate **license information**. For this perform the command 'File'>'Save as...', choose data type 'Internal Library' or 'External Library' and press button **Edit license info....**

In the dialog **Edit Licensing Information** enter the information described below. The license information will be added to the Project Info. When later on the library will be included in a CoDeSys project, the license information can be checked up in the object properties dialog of the library in the library manager.

Dialog: Edit Licensing Information

Common:

Name: Enter a name for the library module which is used to represent it in the *3S Licensing Manager*. This input is mandatory.

Vendor-ID: A manufacturer identifier, depending on the manufacturer specific licensing management tool.

Demo mode: Activate this option if the module should be usable in demo mode that means without any license ID. Enter the number of **days** after which the "demo license" should expire. The number of days will be automatically rounded up to the next number which is divisible by ten (10, 20, 30 ...). If the module should be usable without time limitation, enter "unlimited" (available in the selection list).

Targets: Enter here the target ID(s) of the target system(s) for which the license should be valid. Multiple inputs must be separated by a comma or a semicolon. Multiple IDs can be inserted separated by semicolons resp. as ranges. Example: "12;15-19;21"

Contact:

Licensing via phone: / Licensing per via mail: Insert here the phone number resp. email address of the license provider. These inputs are mandatory.

Optional information:

In the right window you can enter a text referring to the item currently marked in the left window: Description, Manufacturer, Vendor, Pricing information

Please note:

1. It is reasonable to protect a library, which has been provided with licensing information, by a password. If you are going to save the project without password you will be pointed to that by a message box.
 2. The licensing information of a 3S library is stored internally with the library and will be registered on the computer automatically as soon as the library is included in a project. But the license information of modules which are not provided by 3S must be provided in a separate **description file** in compatible XML format, which can be read by the 3S Licensing Manager.
For this also see the separate documentation on the **3S Licensing Manager**.
-

10 APPENDIX

Appendix A: IEC Operators and additional norm extending functions

CoDeSys supports all IEC operators. In contrast with the standard functions (see appendix D, Standard library), these operators are recognized implicitly throughout the project. Besides the IEC operators **CoDeSys also supports the following operators which are not prescribed by the standard**: INDEXOF and SIZEOF (see Arithmetic Operators), ADR and BITADR (see Address Operators).

Operators are used like functions in POU.

Attention: At operations with floating point variables the result depends on the currently used target system!

Arithmetic operators

Bitstring Operators

Bit-Shift Operators

Selection Operators

Comparison Operators

Address Operators

Calling Operators

Type Conversions

Numeric Operators

10.1 Arithmetic Operators...

ADD

Addition of variables of the types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL and LREAL.

Two TIME variables can also be added together resulting in another time (e.g., t#45s + t#50s = t#1m35s)

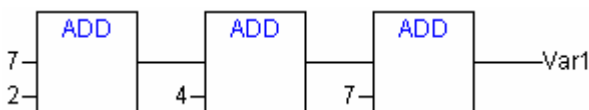
Example in IL:

```
LD 7
ADD 2,4,7
ST Var1
```

Example in ST:

```
var1 := 7+2+4+7;
```

Example in FBD:



MUL

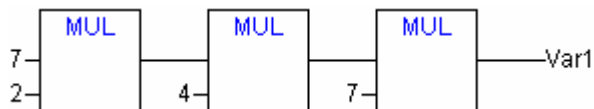
Multiplication of variables of the types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL and LREAL.

Example in IL:

```
LD 7
MUL 2,4,7
ST Var1
```

Example in ST:

```
var1 := 7*2*4*7;
```

Example in FBD:**SUB**

Subtraction of one variable from another of the types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL and LREAL.

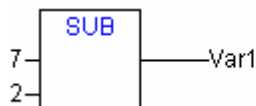
A TIME variable may also be subtracted from another TIME variable resulting in third TIME type variable. Note that negative TIME values are undefined.

Example in IL:

```
LD 7
SUB 2
ST Var1
```

Example in ST:

```
var1 := 7-2;
```

Example in FBD:**DIV**

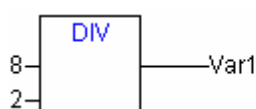
Division of one variable by another of the types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL and LREAL.

Example in IL:

```
LD 8
DIV 2
ST Var1 (* Result is 4 *)
```

Example in ST:

```
var1 := 8/2;
```

Example in FBD:

Note: If you define functions in your project with the names **CheckDivByte**, **CheckDivWord**, **CheckDivDWord** and **CheckDivReal**, you can use them to check the value of the divisor if you use

the operator DIV, for example to avoid a division by 0. The functions must have the above listed names.

Attention: Please regard, that different target systems may behave differently concerning a division by zero !

See in the following an example for the implementation of function CheckDivReal:

Example for the implementation of the function CheckDivReal:

```
FUNCTION CheckDivReal : REAL
VAR INPUT
  divisor:REAL;
END_VAR

IF divisor = 0 THEN
  CheckDivReal:=1;
ELSE
  CheckDivReal:=divisor;
END_IF;
```

Operator DIV uses the output of function CheckDivReal as divisor. In a program like shown in the following example this avoids a division by 0, the divisor (d) is set from 0 to 1. So the result of the division is 799.

```
PROGRAM PLC_PRG
VAR
  erg:REAL;
  v1:REAL:=799;
  d:REAL;
END_VAR

erg:= v1 / d;
```

Attention: The CheckDiv-functions provided by the Check.Lib library just are sample solutions! Before using those library modules check whether they are working in your sense, or implement appropriate functions directly as a POU in your project.

MOD

Modulo Division of one variable by another of the types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT. The result of this function will be the remainder of the division. This result will be a whole number.

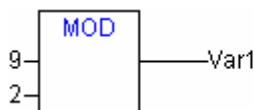
Example in IL:

```
LD 9
MOD 2
ST Var1 (* Result is 1 *)
```

Example in ST:

```
var1 := 9 MOD 2;
```

Example in FBD:

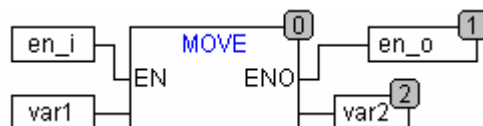


MOVE

Assignment of a variable to another variable of an appropriate type. As MOVE is available as a box in the graphic editors LD, CFC, there the (unlocking) EN/ENO functionality can also be applied on a variable assignment. In the FBD editor this is not possible however.

Example in CFC in conjunction with the EN/ENO function:

Only if en_i is TRUE, var1 will be assigned to var2.



Example in IL:

```
LD ivar1
MOVE
ST ivar2 (* Result: ivar2 gets assigned value of ivar1 *)
```

(! you get the same result with:

```
LD ivar1
ST ivar2 )
```

Example in ST:

```
ivar2 := MOVE(ivar1);
```

(! you get the same result with: ivar2 := ivar1;)

INDEXOF

This function is not prescribed by the standard IEC61131-3.

Perform this function to find the internal index for a POU.

Example in ST:

```
var1 := INDEXOF(POU2);
```

SIZEOF

This function is not prescribed by the standard IEC61131-3.

Perform this function to determine the number of bytes required by the given variable.

Example in IL:

```
arr1:ARRAY[0..4] OF INT;
Var1 INT
LD arr1
SIZEOF
ST Var1 (* Result is 10 *)
```

Example in ST:

```
var1 := SIZEOF(arr1);
```

10.2 Bitstring Operators...

AND

Bitwise AND of bit operands. The operands should be of the type BOOL, BYTE, WORD or DWORD.

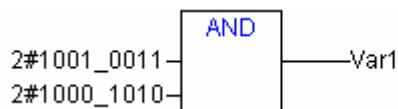
Example in IL:

```
Var1 BYTE
LD 2#1001_0011
AND 2#1000_1010
ST Var1 (* Result is 2#1000_0010 *)
```

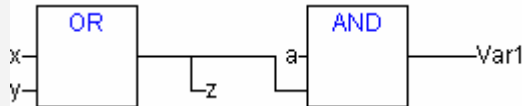
Example in ST:

```
var1 := 2#1001_0011 AND 2#1000_1010
```

Example in FBD:



Note: If you have a program step in the SFC like the following



and if you use 68xxx generators, please note the following: The allocation of the value of the second input variable at the AND operator module to variable z will not be executed ! This is due to the optimized processing in the SFC in case of value FALSE at the input variable.

OR

Bitwise OR of bit operands. The operands should be of the type BOOL, BYTE, WORD or DWORD.

Example in IL:

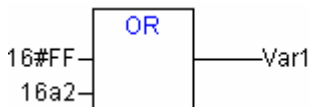
```

var1 :BYTE;
LD 2#1001_0011
OR 2#1000_1010
ST var1 (* Result is 2#1001_1011 *)
  
```

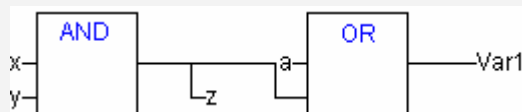
Example in ST:

```
Var1 := 2#1001_0011 OR 2#1000_1010
```

Example in FBD:



Note: If you have a program step in the SFC like the following



and if you use 68xxx generators, please note the following: The allocation of the value of the second input variable at the OR operator module to variable z will not be executed ! This is due to the optimized processing in the SFC in case of value FALSE at the input variable.

XOR

Bitwise XOR of bit operands. The operands should be of the type BOOL, BYTE, WORD or DWORD.

Note: Regard the behaviour of the XOR function in extended form, that means if there are more than 2 inputs. The inputs will be checked in pairs and the particular results will then be compared again in pairs (this complies with the standard, but may not be expected by the user).

Example in IL:

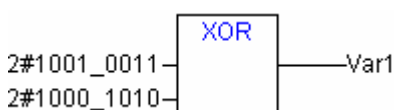
```

Var1 :BYTE;
LD 2#1001_0011
XOR 2#1000_1010
ST Var1 (* Result is 2#0001_1001 *)
  
```

Example in ST:

```
Var1 := 2#1001_0011 XOR 2#1000_1010
```

Example in FBD:



NOT

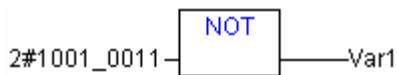
Bitwise NOT of a bit operand. The operand should be of the type BOOL, BYTE, WORD or DWORD.

Example in IL:

```
Var1 :BYTE;
LD 2#1001_0011
NOT
ST Var1 (* Result is 2#0110_1100 *)
```

Example in ST:

```
Var1 := NOT 2#1001_0011
```

Example in FBD:**10.3 Bit-Shift Operators...**

Note: Please note, that the amount of bits, which is regarded for the arithmetic operation, is pretended by the data type of the input variable !. If the input variable is a constant the smallest possible data type is regarded. The data type of the output variable has no effect at all on the arithmetic operation.

SHL

Bitwise left-shift of an operand : erg:= SHL (in, n)

in gets shifted to the left by n bits. If n > data type width, for BYTE, WORD and DWORD will be filled with zeros. But if signed data types are used, like e.g. INT, then an arithmetic shift will be executed in such cases, that means it will be filled with the value of the topmost bit.

Note: See in the following example in hexadecimal notation that you get different results for erg_byte and erg_word depending on the data type of the input variable (BYTE or WORD), although the values of the input variables in_byte and in_word are the same.

Example in ST:

```
PROGRAM shl_st
VAR
in_byte : BYTE:=16#45;
in_word : WORD:=16#45;
erg_byte : BYTE;
erg_word : WORD;
n: BYTE :=2;
END_VAR

erg_byte:=SHL(in_byte,n); (* Result is 16#14 *)
erg_word:=SHL(in_word,n); (* Result is 16#01141 *)
```

Example in FBD:**Example in IL:**

```
LD 16#45
SHL 2
ST erg_byte
```

SHR

Bitwise right-shift of an operand: $erg := SHR(in, n)$

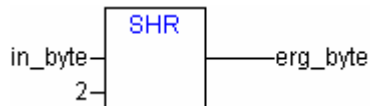
in gets shifted to the right by n bits. If $n >$ data type width, for BYTE, WORD and DWORD will be filled with zeros. But if signed data types are used, like e.g. INT, then an arithmetic shift will be executed in such cases, that means it will be filled with the value of the topmost bit.

See the following example in hexadecimal notation to notice the results of the arithmetic operation depending on the type of the input variable (BYTE or WORD).

Example in ST:

```
PROGRAM shr_st
VAR
in_byte : BYTE:=16#45;
in_word : WORD:=16#45;
erg_byte : BYTE;
erg_word : WORD;
n: BYTE :=2;
END_VAR

erg_byte:=SHR(in_byte,n); (* Result is 11 *)
erg_word:=SHR(in_word,n); (* Result is 0011 *)
```

Example in FBD:**Example in IL:**

```
LD 16#45
SHR 2
ST erg_byte
```

ROL

Bitwise rotation of an operand to the left: $erg := ROL(in, n)$

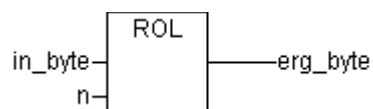
erg, in and n should be of the type BYTE, WORD or DWORD. in will be shifted one bit position to the left n times while the bit that is furthest to the left will be reinserted from the right.

See in the following example in hexadecimal notation that you get different results for erg_byte and erg_word depending on the data type of the input variable (BYTE or WORD), although the values of the input variables in_byte and in_word are the same.

Example in ST:

```
PROGRAM rol_st
VAR
in_byte : BYTE:=16#45;
in_word : WORD:=16#45;
erg_byte : BYTE;
erg_word : WORD;
n: BYTE :=2;
END_VAR

erg_byte:=ROL(in_byte,n); (* Result is 16#15 *)
erg_word:=ROL(in_word,n); (* Result is 16#0114 *)
```

Example in FBD:**Example in IL:**

```
LD 16#45
ROL 2
```

```
ST erg_byte
```

ROR

Bitwise rotation of an operand to the right: $erg = ROR(in, n)$

erg , in and n should be of the type BYTE, WORD or DWORD. in will be shifted one bit position to the right n times while the bit that is furthest to the left will be reinserted from the left.

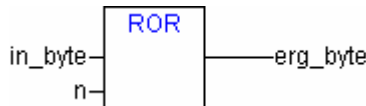
See in the following example in hexadecimal notation that you get different results for erg_byte and erg_word depending on the data type of the input variable (BYTE or WORD), although the values of the input variables in_byte and in_word are the same.

Example in ST:

```
PROGRAM ror_st
VAR
in_byte : BYTE:=16#45;
in_word : WORD:=16#45;
erg_byte : BYTE;
erg_word : WORD;
n: BYTE :=2;
END_VAR

erg_byte:=ROR(in_byte,n); (* Result is 16#51 *)
erg_word:=ROR(in_word,n); (* Result is 16#4011 *)
```

Example in FBD:



Example in IL:

```
LD 16#45
ROR 2
ST erg_byte
```

10.4 Selection Operators

All selection operations can also be performed with variables. For purposes of clarity we will limit our examples to the following which use constants as operators.

SEL

Binary Selection.

```
OUT := SEL(G, IN0, IN1) means:
OUT := IN0 if G=FALSE;
OUT := IN1 if G=TRUE.
```

$IN0$, $IN1$ and OUT can be any type of variable, G must be BOOL. The result of the selection is $IN0$ if G is FALSE, $IN1$ if G is TRUE.

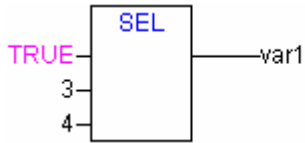
Example in IL:

```
LD TRUE
SEL 3,4 (* IN0 = 3, IN1 =4 *)
ST Var1 (* Result is 4 *)
LD FALSE
SEL 3,4
ST Var1 (* Result is 3 *)
```

Example in ST:

```
Var1:=SEL(TRUE,3,4); (* Result is 4 *)
```

Example in FBD:



Note: Note that an expression occurring ahead of IN1 or IN2 will not be processed if IN0 is TRUE.

MAX

Maximum function. Returns the greater of the two values.

```
OUT := MAX(IN0, IN1)
```

IN0, IN1 and OUT can be any type of variable.

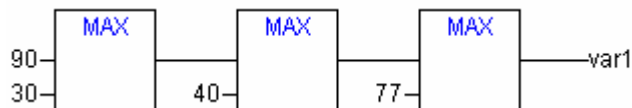
Example in IL:

```
LD 90
MAX 30
MAY 40
MAX 77
ST Var1 (* Result is 90 *)
```

Example in ST:

```
Var1:=MAX(30,40); (* Result is 40 *)
Var1:=MAX(40,MAX(90,30)); (* Result is 90 *)
```

Example in FBD:



MIN

Minimum function. Returns the lesser of the two values.

```
OUT := MIN(IN0, IN1)
```

IN0, IN1 and OUT can be any type of variable.

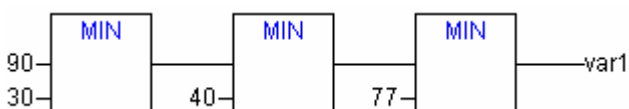
Example in IL:

```
LD 90
MIN 30
MIN 40
MIN 77
ST Var1 (* Result is 30 *)
```

Example in ST:

```
Var1:=MIN(90,30); (* Result is 30 *)
Var1:=MIN(MIN(90,30),40); (* Result is 30 *)
```

Example in FBD:



LIMIT

Limiting

`OUT := LIMIT(Min, IN, Max)` means:

`OUT := MIN (MAX (IN, Min), Max)`

Max is the upper and Min the lower limit for the result. Should the value IN exceed the upper limit Max, LIMIT will return Max. Should IN fall below Min, the result will be Min.

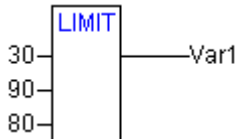
IN and OUT can be any type of variable.

Example in IL:

```
LD 90
LIMIT 30,80
ST Var1 (* Result is 80 *)
```

Example in ST:

```
Var1:=LIMIT(30,90,80); (* Result is 80 *);
```

Beispiel in FBD:**MUX**

Multiplexer

`OUT := MUX(K, IN0, ..., INn)` means:

`OUT := INK.`

IN0, ...,INn and OUT can be any type of variable. K must be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT or UDINT. MUX selects the Kth value from among a group of values.

Example in IL:

```
LD 0
MUX 30,40,50,60,70,80
ST Var1 (* Result is 30 *)
```

Example in ST:

```
Var1:=MUX(0,30,40,50,60,70,80); (* Result is 30 *);
```

Please note: An expression occurring ahead of an input other than INK will not be processed to save run time !
Only in simulation mode all expressions will be executed.

10.5 Comparison Operators...

GT

Greater than

A Boolean operator which returns the value TRUE when the value of the first operand is greater than that of the second. The operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME and STRING.

Example in IL:

```
LD 20
GT 30
ST Var1 (* Result is FALSE *)
```

Example in ST:

```
VAR1 := 20 > 30 > 40 > 50 > 60 > 70;
```

Example in FBD:



LT

Less than

A Boolean operator that returns the value TRUE when the value of the first operand is less than that of the second. The operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME and STRING.

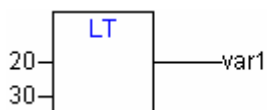
Example in IL:

```
LD 20
LT 30
ST Var1 (* Result is TRUE *)
```

Example in ST:

```
VAR1 := 20 < 30;
```

Example in FBD:



LE

Less than or equal to

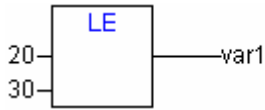
A Boolean operator that returns the value TRUE when the value of the first operand is less than or equal to that of the second. The operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME and STRING.

Example in IL:

```
LD 20
LE 30
ST Var1 (* Result is TRUE *)
```


Example in ST:

```
VAR1 := 20 <= 30;
```

Example in FBD:**GE**

Greater than or equal to

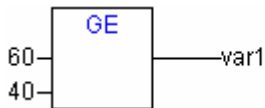
A Boolean operator that returns the value TRUE when the value of the first operand is greater than or equal to that of the second. The operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME and STRING.

Example in IL:

```
LD 60
GE 40
ST Var1 (* Result is TRUE *)
```

Example in ST:

```
VAR1 := 60 >= 40;
```

Example in FBD:**EQ**

Equal to

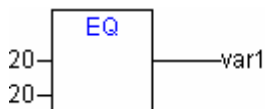
A Boolean operator that returns the value TRUE when the operands are equal. The operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME and STRING.

Example in IL:

```
LD 40
EQ 40
ST Var1 (* Result is TRUE *)
```

Example in ST:

```
VAR1 := 40 = 40;
```

Example in FBD:**NE**

Not equal to

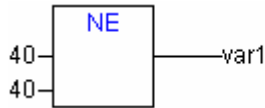
A Boolean operator that returns that value TRUE when the operands are not equal. The operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME and STRING.

Example in IL:

```
LD 40
NE 40
ST Var1 (* Result is FALSE *)
```

Example in ST:

```
VAR1 := 40 <> 40;
```

Example in FBD:

10.6 Address Operators...

Attention: After an Online Change there might be changes concerning the data on certain addresses. Please regard this in case of using pointers on addresses.

ADR

Address Function not prescribed by the standard IEC61131-3.

ADR returns the address of its argument in a DWORD. This address can be sent to manufacturing functions to be treated as a pointer or it can be assigned to a pointer within the project.

```
dwVar:=ADR(bVAR);
```

Example in IL:

```
LD bVar
ADR
ST dwVar
man_fun1
```

ADRINST

Address function, not prescribed by the standard IEC61131-3.

ADRINST within a function block instance returns the address of the instance in a DWORD. This address then can be sent to functions and be treated there as a pointer or it can be assigned to a pointer within the project.

Examples in ST (within a function block instance):

```
dvar:=ADRINST(); (* Write address of the instance on variable dvar *)
fun(a:=ADRINST()); (* Give instance address to input a of function fun *)
```

Examples in IL:

```
ADRINST
ST dvar

ADRINST
fun
```

BITADR

Address function, not prescribed by the standard IEC61131-3.

BITADR returns the bit offset within the segment in a DWORD. Regard that the offset value depends on whether the option byte addressing in the target settings is activated or not.

```
VAR
var1 AT %IX2.3:BOOL;
bitoffset: DWORD;

END_VAR
```

Example in ST:

```
bitoffset:=BITADR(var1); (* Result if byte addressing=TRUE: 19, if byte
addressing=FALSE: 35 *)
```

Example in IL:

```
LD Var1
BITADR
ST Var2
```

Content Operator

A pointer can be dereferenced by adding the content operator "^" after the pointer identifier.

Example in ST:

```
pt:POINTER TO INT;
var_int1:INT;
var_int2:INT;
pt := ADR(var_int1);
var_int2:=pt^;
```

10.7 Calling Operators...

CAL

Calling a function block or a program

Use CAL in IL to call up a function block instance. The variables that will serve as the input variables are placed in parentheses right after the name of the function block instance.

Example:

Calling up the instance *Inst* from a function block where input variables *Par1* and *Par2* are 0 and TRUE respectively.

```
CAL INST(PAR1 := 0, PAR2 := TRUE)
```

10.8 Type Conversions...

Its is forbidden to implicitly convert from a "larger" type to a "smaller" type (for example from INT to BYTE or from DINT to WORD). Special type conversions are required if one wants to do this. One can basically convert from any elementary type to any other elementary type.

Syntax:

```
<elem.Typ1>_TO_<elem.Typ2>
```

Please regard that at ...TO_STRING conversions the string is generated left-justified. If it is defined to short, it will be cut from the right side.

BOOL_TO Conversions

Conversion from type BOOL to any other type:

For number types the result is 1, when the operand is TRUE, and 0, when the operand is FALSE.

For the STRING type the result is ,TRUE' or ,FALSE'.

Examples in IL:

```
LD TRUE (*Result is 1 *)
BOOL_TO_INT
ST i
```

Appendix A: - IEC Operators and additional norm extending functions

```

LD TRUE                                (*Result is 'TRUE' *)
BOOL_TO_STRING
ST str

LD TRUE                                (*Result is T#1ms *)
BOOL_TO_TIME
ST t

LD TRUE                                (*Result is TOD#00:00:00.001 *)
BOOL_TO_TOD
ST

LD FALSE                               (*Result is D#1970-01-01 *)
BOOL_TO_DATE
ST dat

LD TRUE                                (*Result is DT#1970-01-01-00:00:01 *)
BOOL_TO_DT
ST dandt

```

Examples in ST:

```

i:=BOOL_TO_INT(TRUE);                 (* Result is 1 *)

str:=BOOL_TO_STRING(TRUE);            (* Result is "TRUE" *)

t:=BOOL_TO_TIME(TRUE);                (* Result is T#1ms *)

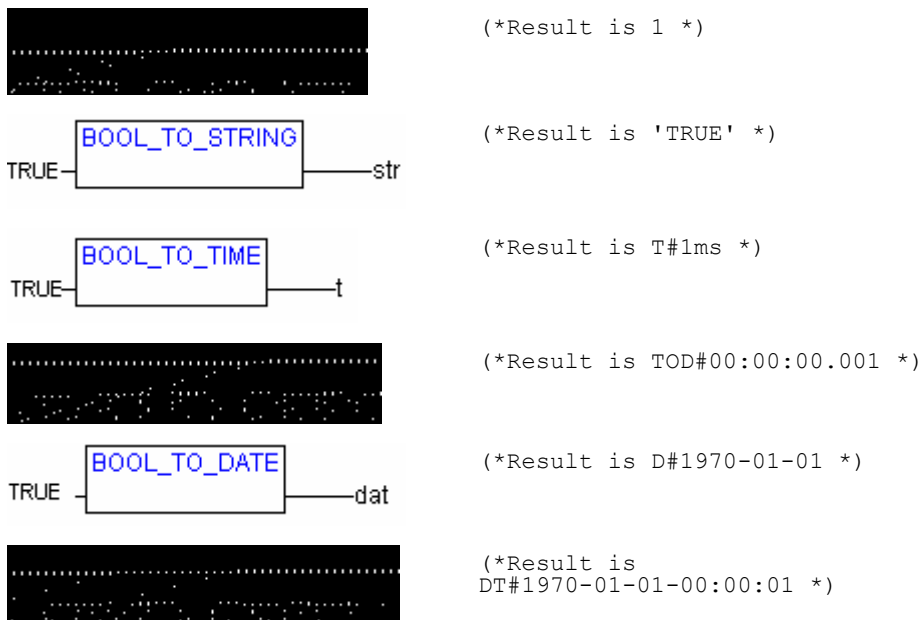
tof:=BOOL_TO_TOD(TRUE);               (* Result is TOD#00:00:00.001 *)

dat:=BOOL_TO_DATE(FALSE);             (* Result is D#1970 *)

dandt:=BOOL_TO_DT(TRUE);              (* Result is
DT#1970-01-01-00:00:01 *)

```

Examples in FBD:



TO_BOOL Conversions

Conversion from another variable type to BOOL:

The result is TRUE when the operand is not equal to 0. The result is FALSE when the operand is equal to 0.

The result is true for STRING type variables when the operand is "TRUE", otherwise the result is FALSE.

Examples in IL:

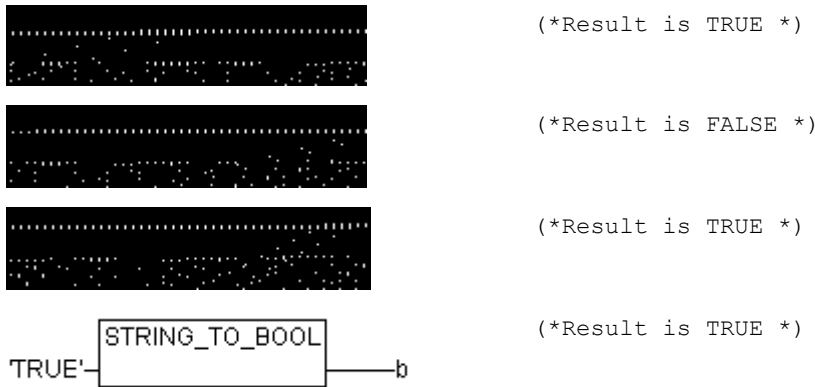
```
LD 213 (*Result is TRUE *)
BYTE_TO_BOOL
ST b

LD 0 (*Result is FALSE *)
INT_TO_BOOL
ST b

LD T#5ms (*Result is TRUE *)
TIME_TO_BOOL
ST b

LD 'TRUE' (*Result is TRUE *)
STRING_TO_BOOL
ST b
```

Examples in FBD:



Examples in St:

```
b := BYTE_TO_BOOL(2#11010101); (* Result is TRUE *)
b := INT_TO_BOOL(0); (* Result is FALSE *)
b := TIME_TO_BOOL(T#5ms); (* Result is TRUE *)
b := STRING_TO_BOOL('TRUE'); (* Result is TRUE *)
```

Conversion between Integral Number Types

Conversion from an integral number type to another number type:

When you perform a type conversion from a larger to a smaller type, you risk losing some information. If the number you are converting exceeds the range limit, the first bytes for the number will be ignored.

Example in ST:

```
si := INT_TO_SINT(4223); (* Result is 127 *)
```

If you save the integer 4223 (16#107f represented hexadecimally) as a SINT variable, it will appear as 127 (16#7f represented hexadecimally).

Example in IL:

```
LD 2
INT_TO_REAL
MUL
```

Example in FBD:**REAL_TO-/ LREAL_TO Conversions**

Converting from the variable type REAL or LREAL to a different type:

The value will be rounded up or down to the nearest whole number and converted into the new variable type. Exceptions to this are the variable types STRING, BOOL, REAL and LREAL.

Please regard at a conversion to type STRING that the total number of digits is limited to 16. If the (L)REAL-number has more digits, then the sixteenth will be rounded. If the length of the STRING is defined to short, it will be cut beginning from the right end.

When you perform a type conversion from a larger to a smaller type, you risk losing some information.

Example in ST:

```
i := REAL_TO_INT(1.5); (* Result is 2 *)
j := REAL_TO_INT(1.4); (* Result is 1 *)
i := REAL_TO_INT(-1.5); (* Result is -2 *)
j := REAL_TO_INT(-1.4); (* Result is -1 *)
```

Example in IL:

```
LD 2.7
REAL_TO_INT
GE %MW8
```

Example in FBD:**TIME_TO/TIME_OF_DAY Conversions**

Converting from the variable type TIME or TIME_OF_DAY to a different type:

The time will be stored internally in a DWORD in milliseconds (beginning with 12:00 A.M. for the TIME_OF_DAY variable). This value will then be converted.

When you perform a type conversion from a larger to a smaller type, you risk losing some information

For the STRING type variable, the result is a time constant.

Examples in IL:

```
LD T#12ms (*Result is 'T#12ms' *)
TIME_TO_STRING
ST str

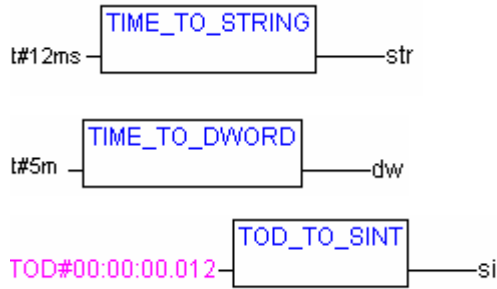
LD T#300000ms (*Result is 300000 *)
TIME_TO_DWORD
ST dw

LD TOD#00:00:00.012 (*Result is 12 *)
TOD_TO_SINT
ST si
```

Examples in ST:

```
str :=TIME_TO_STRING(T#12ms); (* Result is T#12ms *)
dw:=TIME_TO_DWORD(T#5m); (* Result is 300000 *)
si:=TOD_TO_SINT(TOD#00:00:00.012); (* Result is 12 *)
```

Examples in FBD:



DATE_TO/DT_TO Conversions

Converting from the variable type DATE or DATE_AND_TIME to a different type:

The date will be stored internally in a DWORD in seconds since Jan. 1, 1970. This value will then be converted.

When you perform a type conversion from a larger to a smaller type, you risk losing some information

For STRING type variables, the result is the date constant.

Examples in IL:

```
LD D#1970-01-01          (* Result is FALSE *)
DATE_TO_BOOL
ST b

LD D#1970-01-15        (* Result is 29952 *)
DATE_TO_INT
ST i

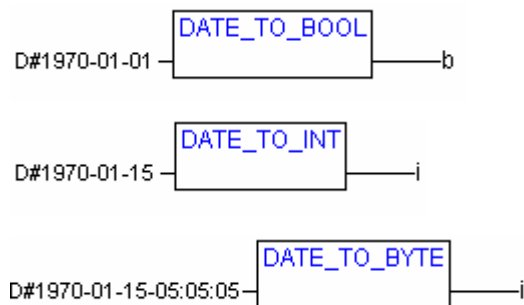
LD DT#1970-01-15-05:05:05 (* Result is 129 *)
DT_TO_BYTE
ST byt

LD DT#1998-02-13-14:20 (* Result is 'DT#1998-02-13-14:20' *)
DT_TO_STRING
ST str
```

Examples in ST:

```
b :=DATE_TO_BOOL(D#1970-01-01);          (* Result is FALSE *)
i :=DATE_TO_INT(D#1970-01-15);          (* Result is 29952 *)
byt :=DT_TO_BYTE(DT#1970-01-15-05:05:05); (* Result is 129 *)
str:=DT_TO_STRING(DT#1998-02-13-14:20); (* Result is
'DT#1998-02-13-14:20' *)
```

Examples in FBD:





STRING_TO Conversions

Converting from the variable type STRING to a different type:

The operand from the STRING type variable must contain a value that is valid in the target variable type, otherwise the result will be 0.

Examples in IL:

```

LD 'TRUE' (* Result is TRUE *)
STRING_TO_BOOL
ST b

LD 'abc34' (* Result is 0 *)
STRING_TO_WORD
ST w

LD 't#127ms' (* Result is T#127ms *)
STRING_TO_TIME
ST t
  
```

Examples in ST:

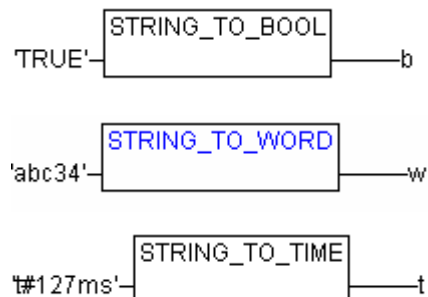
```

b :=STRING_TO_BOOL('TRUE'); (* Result is TRUE *)

w :=STRING_TO_WORD('abc34'); (* Result is 0 *)

t :=STRING_TO_TIME('T#127ms'); (* Result is T#127ms *)
  
```

Examples in FBD:



TRUNC

Converting from REAL to INT. The whole number portion of the value will be used.

When you perform a type conversion from a larger to a smaller type, you risk losing some information.

Example in IL:

```

LD 2.7
TRUNC
GE %MW8
  
```

Examples in ST:

```

i:=TRUNC(1.9); (* Result is 1 *)
i:=TRUNC(-1.4); (* Result is -1 *)
  
```


10.9 Numeric Operators...

ABS

Returns the absolute value of a number. ABS(-2) equals 2.

The following type combinations for input and output variables are possible:

IN	OUT
INT	INT, REAL, WORD, DWORD, DINT
REAL	REAL
BYTE	INT, REAL, BYTE, WORD, DWORD, DINT
WORD	INT, REAL, WORD, DWORD, DINT
DWORD	REAL, DWORD, DINT
SINT	REAL
USINT	REAL
UINT	INT, REAL, WORD, DWORD, DINT, UDINT, UINT
DINT	REAL, DWORD, DINT
UDINT	REAL, DWORD, DINT, UDINT

Example in IL:

```
LD -2
ABS
ST i (* Result is 2 *)
```

Example in ST:

```
i:=ABS(-2);
```

Example in FBD:



SQRT

Returns the square root of a number.

IN can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, OUT must be type REAL.

Example in IL:

```
LD 16
SQRT
ST q (* Result is 4 *)
```

Example in ST:

```
q:=SQRT(16);
```

Example in FBD:



LN

Returns the natural logarithm of a number.

IN can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, OUT must be type REAL.

Example in IL:

```
LD 45
LN
ST q (* Result is 3.80666 *)
```

Example in ST:

```
q:=LN(45);
```

Example in FBD:**LOG**

Returns the logarithm of a number in base 10.

IN can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, OUT must be type REAL.

Example in IL:

```
LD 314.5
LOG
ST q (* Result is 2.49762 *)
```

Example in ST:

```
q:=LOG(314.5);
```

Example in FBD:**EXP**

Returns the exponential function.

IN can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, OUT must be type REAL.

Example in IL:

```
LD 2
EXP
ST q (* Result is 7.389056099 *)
```

Example in ST:

```
q:=EXP(2);
```

Example in FBD:

SIN

Returns the sine of a number.

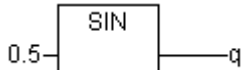
The input value IN is calculated in arch minutes. It can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. OUT must be type REAL.

Example in IL:

```
LD 0.5
SIN
ST q (* Result is 0.479426 *)
```

Example in ST:

```
q:=SIN(0.5);
```

Example in FBD:**COS**

Returns the cosine of number. The result is calculated in arch minutes.

IN can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, OUT must be type Typ REAL.

Example in IL:

```
LD 0.5
COS
ST q (* Result is 0.877583 *)
```

Example in ST:

```
q:=COS(0.5);
```

Example in FBD:**TAN**

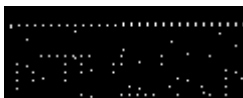
Returns the tangent of a number. The value is calculated in arch minutes. IN can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, OUT must be type REAL.

Example in IL:

```
LD 0.5
TAN
ST q (* Result is 0.546302 *)
```

Example in ST:

```
q:=TAN(0.5);
```

Example in FBD:

ASIN

Returns the arc sine (inverse function of sine) of a number. .

IN can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, OUT must be type REAL.

Example in IL:

```
LD 0.5
ASIN
ST q (* Result is 0.523599 *)
```

Example in ST:

```
q:=ASIN(0.5);
```

Example in FBD:**ACOS**

Returns the arc cosine (inverse function of cosine) of a number. The value is calculated in arch minutes.

IN can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, OUT must be type REAL.

Example in IL:

```
LD 0.5
ACOS
ST q (* Result is 1.0472 *)
```

Example in ST:

```
q:=ACOS(0.5);
```

Example in FBD:**ATAN**

Returns the arc tangent (inverse function of tangent) of a number. IN can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. The result OUT is calculated in arch minutes and must be type REAL.

Example in IL:

```
LD 0.5
ATAN
ST q (* Result is 0.463648 *)
```

Example in ST:

```
q:=ATAN(0.5);
```

Example in FBD:

EXPT

Exponentiation of a variable with another variable:

```
OUT = IN1IN2.
```

IN1 and IN2 can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT, OUT must be type REAL.

Example in IL:

```
LD 7
EXPT 2
ST var1 (* Result is 49 *)
```

Example in ST:

```
var1 := EXPT(7,2);
```

Example in FBD:

10.10 Initialization Operator

INI Operator

The INI operator can be used to initialize retain variables which are provided by a function block instance used in the POU.

The operator must be assigned to a boolean variable.

Syntax: <bool-Variable> := INI(<FB-instance, TRUE|FALSE)

If the second parameter of the operator is set to TRUE, all retain variables defined in the function block FB will be initialized.

Example in ST: fbinst is the instance of function block fb, in which a retain variable retvar is defined.

Declaration in POU:

```
fbinst:fb;
b:bool;
```

Implementation part:

```
b := INI(fbinst, TRUE);
ivar:=fbinst.retvar (* => retvar gets initialized *)
```

Example of operator call in IL:

```
LD fbinst
INI TRUE
ST b
```

Example of operator call in FBD:

Appendix B: Operands in CoDeSys

In CoDeSys Constants, variables, addresses and possibly function calls can appear as operands.

10.11 Constants

BOOL Constants

BOOL constants are the logical values TRUE and FALSE.

TIME Constants

TIME constants can be declared in **CoDeSys**. These are generally used to operate the timer in the standard library. A TIME constant is always made up of an initial "t" or "T" (or "time" or "TIME" spelled out) and a number sign "#".

This is followed by the actual time declaration which can include days (identified by "d"), hours (identified by "h"), minutes (identified by "m"), seconds (identified by "s") and milliseconds (identified by "ms"). Please note that the time entries must be given in this order according to length (d before h before m before s before m before ms) but you are not required to include all time increments.

Examples of correct TIME constants in a ST assignment:

```
TIME1 := T#14ms;
```

```
TIME1 := T#100S12ms; (*The highest component may be allowed to exceed its limit*)
```

```
TIME1 := t#12h34m15s;
```

the following would be incorrect:

```
TIME1 := t#5m68s; (*limit exceeded in a lower component*)
```

```
TIME1 := 15ms; (*T# is missing*)
```

```
TIME1 := t#4ms13d; (*Incorrect order of entries*)
```

DATE Constants

These constants can be used to enter dates. A DATE constant is declared beginning with a "d", "D", "DATE" or "date" followed by "#". You can then enter any date with format Year-Month-Day.

Examples:

```
DATE#1996-05-06
```

```
d#1972-03-29
```

(see also Chapter 10.15 Data types, Time Data Types)

TIME_OF_DAY Constants

Use this type of constant to store times of the day. A TIME_OF_DAY declaration begins with "tod#", "TOD#", "TIME_OF_DAY#" or "time_of_day#" followed by a time with the format: Hour:Minute:Second. You can enter seconds as real numbers or you can enter fractions of a second.

Examples:

```
TIME_OF_DAY#15:36:30.123
```

```
tod#00:00:00
```

(see also Chapter 10.15 Data types, Time Data Types)

DATE_AND_TIME Constants

Date constants and the time of day can also be combined to form so-called DATE_AND_TIME constants. DATE_AND_TIME constants begin with "dt#", "DT#", "DATE_AND_TIME#" or "date_and_time#". Place a hyphen after the date followed by the time.

Examples:

```
DATE_AND_TIME#1996-05-06-15:36:30
```

```
dt#1972-03-29-00:00:00
```

(see also Chapter 10.15 Data types, Time Data Types)

Number Constants

Number values can appear as binary numbers, octal numbers, decimal numbers and hexadecimal numbers. If an integer value is not a decimal number, you must write its base followed by the number sign (#) in front of the integer constant. The values for the numbers 10-15 in hexadecimal numbers will be represented as always by the letters A-F.

You may include the underscore character within the number.

Examples:

```
14          (decimal number)
```

```
2#1001_0011 (dual number)
```

```
8#67       (octal number)
```

```
16#A       (hexadecimal number)
```

These number values can be from the variable types BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL or LREAL.

Implicit conversions from "larger" to "smaller" variable types are not permitted. This means that a DINT variable cannot simply be used as an INT variable. You must use the type conversion.

REAL/LREAL Constants

REAL and LREAL constants can be given as decimal fractions and represented exponentially. Use the standard American format with the decimal point to do this.

Example:

```
7.4 instead of 7,4
```

```
1.64e+009 instead of 1,64e+009
```

STRING Constants

A string is a sequence of characters. STRING constants are preceded and followed by single quotation marks. You may also enter blank spaces and special characters (umlauts for instance). They will be treated just like all other characters.

In character sequences, the combination of the dollar sign (\$) followed by two hexadecimal numbers is interpreted as a hexadecimal representation of the eight bit character code. In addition, the combination of two characters that begin with the dollar sign are interpreted as shown below when they appear in a character sequence:

```
$$          Dollar signs
$'          Single quotation mark
$L or $!    Line feed
$N or $n    New line
$P or $p    Page feed
```

\$R or \$r Line break

\$T or \$t Tab

Examples:

```
'w1Wüß?'
' Abby and Craig '
':-)'
```

Typed Literals

Basically, in using IEC constants, the smallest possible data type will be used. If another data type must be used, this can be achieved with the help of typed literals without the necessity of explicitly declaring the constants. For this, the constant will be provided with a prefix which determines the type.

This is written as follows: <Type>#<Literal>

<Type> specifies the desired data type; possible entries are: BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, LREAL. The type must be written in uppercase letters.

<Literal> specifies the constant. The data entered must fit within the data type specified under <Type>.

Example:

```
var1:=DINT#34;
```

If the constant can not be converted to the target type without data loss, an error message is issued:

Typed literals can be used wherever normal constants can be used.

10.12 Variables

Variables can be declared either locally in the declaration part of a POU or in a global variable list.

Please regard: In a project you can define a local variable which has the same name like a global variable. In this case within a POU the locally defined variable will be used. It is not allowed however to name two global variables identically. For example you will get a compiler error, if you have defined a variable "var1" in the PLC Configuration as well as in a global variables list.

The variable identifier may not contain any blank spaces or special characters, may not be declared more than once and cannot be the same as any of the keywords. Capitalization is not recognized which means that VAR1, Var1, and var1 are all the same variable. The underscore character is recognized in identifiers (e.g., "A_BCD" and "AB_CD" are considered two different identifiers). An identifier may not have more than one underscore character in a row. The length of the identifier, as well as the meaningful part of it, are unlimited.

Variables can be used anywhere the declared type allows for them.

You can access available variables through the Input Assistant.

System Flags

System flags are implicitly declared variables that are different on each specific PLC. To find out which system flags are available in your system, use the command **'Insert' 'Operand'** An Input Assistant dialog box pops up, select the category **System Variable**.

Accessing variables for arrays, structures and POUs.

Two-dimensional array components can be accessed using the following syntax:

```
<Fieldname>[Index1, Index2]
```

Structure variables can be accessed using the following syntax:

```
<Structurename>.<Variablennamen>
```


Function block and program variables can be accessed using the following syntax:

```
<Functionblockname>.<Variablename>
```

Addressing bits in variables

In integer variables individual bits can be accessed. For this, the index of the bit to be addressed is appended to the variable, separated by a dot. The bit-index can be given by any constant. Indexing is 0-based.

Please regard: Bitaccess in Direct variables is not allowed.

Example:

```
a : INT;
b : BOOL;
..
a.2 := b;
```

The third bit of the variable a will be set to the value of the variable b.

If the index is greater than the bit width of the variable, the following error message is issued: Index '<n>' outside the valid range for variable '<var>'!

Bit addressing is possible with the following variable types: SINT, INT, DINT, USINT, UINT, UDINT, BYTE, WORD, DWORD.

If the variable type does not allow it, the following error message is issued: "Invalid data type '<type>' for direct indexing"

A bit access must not be assigned to a VAR_IN_OUT variable!

Bitaccess via a global constant:

If you have declared a global constant, which defines the bit-index, you can use this constant for a bitaccess.

Please regard: The project option 'Replace constants' (category Build) must be activated !

See in the following examples for a bitaccess on a variable resp. a structure variable:

Declaration in global variables list for both examples:

Variable enable defines which bit should be accessed:

```
VAR_GLOBAL CONSTANT
enable:int:=2;
END_VAR
```

Example 1, Bitaccess on an integer variable:

Declaration in POU:

```
VAR
xxx:int;
END_VAR
```

Bitaccess:

```
xxx.enable:=true; -> the third second bit in variable xxx will be set TRUE
```

Example 2, Bitaccess on an integer structure component:

Declaration of structure stru1:

```
TYPE stru1 :
STRUCT
bvar:BOOL;
rvar:REAL;
wvar:WORD;
```

```
{bitaccess enable 42 'Start drive'}
END_STRUCT
END_TYPE
```

Declaration in POU:

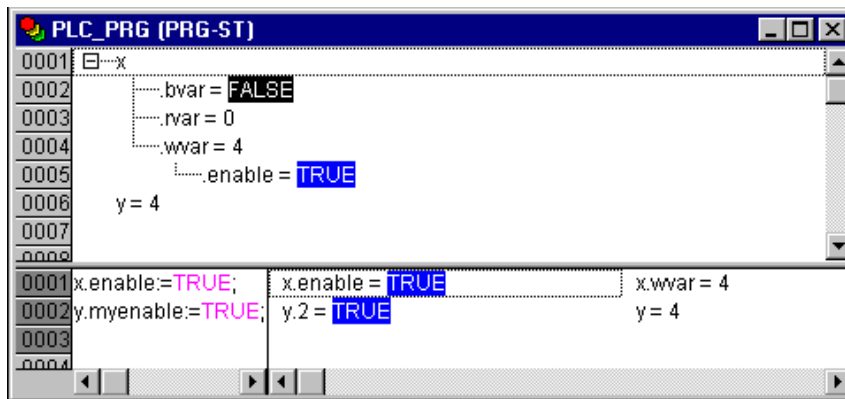
```
VAR
x:struc1;
END_VAR
```

Bitaccess:

```
x.enable:=true;
```

This will set TRUE the 42. bit in variable x. Since bvar has 8 bits and rvar has 32 bits, the bitaccess will be done on the second bit of variable wvar, which as a result will get value 4.

Attention: If a variable, which does a bitaccess on a structure variable with the aid of a global constant, should be displayed correctly in the input assistant, at monitoring in the declaration window and in the "Intellisense function", please use **pragma {bitaccess}** as shown in the example. Then in addition you get displayed the global constant beyond the respective structure variable during monitoring in the declaration window:



10.13 Addresses

Address

The direct display of individual memory locations is done through the use of special character sequences. These sequences are a concatenation of the percent sign "%", a range prefix, a prefix for the size and one or more natural numbers separated by blank spaces.

The following range prefixes are supported:

- I Input
- Q Output
- M Memory location

The following size prefixes are supported:

- X Single bit
- None Single bit
- B Byte (8 Bits)
- W Word (16 Bits)
- D Double word (32 Bits)

Examples:

- %QX7.5 and Output bit 7.5
- %Q7.5
- %IW215 Input word 215
- %QB7 Output byte 7
- %MD48 Double word in memory position 48 in the memory location.
- %IW2.5.7.1 depending on the PLC Configuration

The current PLC Configuration for the program determines whether or not an address is valid.

Note: Boolean values will be allocated bitwise, if no explicit single-bit address is specified. Example: A change in the value of varbool1 AT %QW0 affects the range from QX0.0 to QX0.7.

see also Chapter Appendix A: IEC Operators and additional norm extending functions, address operators

Memory location

You can use any supported size to access the memory location.

For example, the address %MD48 would address bytes numbers 192, 193, 194, and 195 in the memory location area (48 * 4 = 192). The number of the first byte is 0.

You can access words, bytes and even bits in the same way: the address %MX5.0 allows you to access the first bit in the fifth word (Bits are generally saved wordwise).

see also Appendix A: IEC Operators and additional norm extending functions, address operators

10.14 Functions

In ST a function call can also appear as an operand.

Example:

```
Result := Fct(7) + 3;
```

TIME()-Function

This function returns the time (based on milliseconds) which has been passed since the system was started.

The data type is TIME.

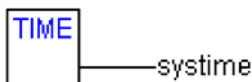
Example in IL:

```
TIME
ST systime (* Result e.g.: T#35m11s342ms *)
```

Example in ST:

```
systime:=TIME();
```

Example in FBD:



Appendix C: Data types in CoDeSys

10.15 Standard data types

You can use standard data types and user-defined data types when programming. Each identifier is assigned to a data type which dictates how much memory space will be reserved and what type of values it stores.

BOOL

BOOL type variables may be given the values TRUE and FALSE. 8 bits of memory space will be reserved.

see also chapter 10.11, Operands in CoDeSys, BOOL constants

Integer Data Types

BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, and UDINT are all integer data types

Each of the different number types covers a different range of values. The following range limitations apply to the integer data types:

Type	Lower limit	Upper limit	Memory space
BYTE	0	255	8 Bit
WORD	0	65535	16 Bit
DWORD	0	4294967295	32 Bit
SINT:	-128	127	8 Bit
USINT:	0	255	8 Bit
INT:	-32768	32767	16 Bit
UINT:	0	65535	16 Bit
DINT:	-2147483648	2147483647	32 Bit
UDINT:	0	4294967295	32 Bit

As a result when larger types are converted to smaller types, information may be lost.

see also Chapter 10.11, Operands in CoDeSys, Number constants

REAL / LREAL

REAL and **LREAL** are so-called floating-point types. They are required to represent rational numbers. 32 bits of memory space is reserved for REAL and 64 bits for LREAL.

Valid values for REAL: 1.175494351e-38 to 3.402823466e+38

Valid values for LREAL: 2.2250738585072014e-308 to 1.7976931348623158e+308

see also Chapter 10.11, REAL-/LREAL constants

STRING

A **STRING** type variable can contain any string of characters. The size entry in the declaration determines how much memory space should be reserved for the variable. It refers to the number of characters in the string and can be placed in parentheses or square brackets. If no size specification is given, the default size of 80 characters will be used.

The string length basically is not limited in CoDeSys, but string functions only can process strings of 1 - 255 characters !

Example of a String Declaration with 35 characters:

```
str:STRING(35):='This is a String';
```

see also Chapter 10.11, Operands in CoDeSys, STRING constants

Time Data Types

The data types **TIME**, **TIME_OF_DAY** (abb. **TOD**), **DATE** and **DATE_AND_TIME** (abb. **DT**) are handled internally like DWORD.

Time is given in milliseconds in TIME and TOD, time in TOD begins at 12:00 A.M.

Time is given in seconds in DATE and DT beginning with January 1, 1970 at 12:00 A.M.

See in the following the time data formats used to assign values for time constants:

TIME constants:

always made up of an initial "t" or "T" (or "time" or "TIME" spelled out) and a number sign "#".

This is followed by the actual time declaration which can include days (identified by "d"), hours (identified by "h"), minutes (identified by "m"), seconds (identified by "s") and milliseconds (identified by "ms"). Please note that the time entries must be given in this order according to length (d before h before m before s before m before ms) but you are not required to include all time increments.

Maximum value: 49d17h2m47s295ms (4194967295 ms)

Examples of correct TIME constants in a ST assignment:

```
TIME1 := T#14ms;
```

```
TIME1 := T#100S12ms;      (*The highest component may be allowed to exceed its limit*)
```

```
TIME1 := t#12h34m15s;
```

the following would be incorrect:

```
TIME1 := t#5m68s;      (*limit exceeded in a lower component*)
```

```
TIME1 := 15ms;        (*T# is missing*)
```

```
TIME1 := t#4ms13d;    (*Incorrect order of entries*)
```

DATE Constants:

beginning with a "d", "D", "DATE" or "date" followed by "#". You can then enter any date with format Year-Month-Day. Possible values: 1970-00-00 to 2106-02-06.

Examples:

```
DATE#1996-05-06
```

```
d#1972-03-29
```

TIME_OF_DAY Constants, for storing times of the day:

begin with "tod#", "TOD#", "TIME_OF_DAY#" or "time_of_day#" followed by a time with the format: Hour:Minute:Second. Seconds can be entered as real numbers or you can enter fractions of a second. Possible values: 00:00:00 bis 23:59:59.999.

Examples:

```
TIME_OF_DAY#15:36:30.123
```

```
tod#00:00:00
```

DATE_AND_TIME Constants, combination of date and the time of day:

begin with "dt#", "DT#", "DATE_AND_TIME#" or "date_and_time#". Place a hyphen after the date followed by the time. Possible values: 1970-00-00-00:00:00 to 2106-02-06-06:28:15.

Examples:

```
DATE_AND_TIME#1996-05-06-15:36:30
dt#1972-03-29-00:00:00
```

10.16 Defined data types

ARRAY

One-, two-, and three-dimensional fields (arrays) are supported as elementary data types. Arrays can be defined both in the declaration part of a POU and in the global variable lists. Maximum 9 dimensions may result from nesting of arrays ("ARRAY[0..2] OF ARRAY[0..3] OF ...").

Syntax:

```
<Field_Name>:ARRAY [<l11>..

```

l1, l2, l3 identify the lower limit of the field range; ul1, ul2 and ul3 identify the upper limit. The limit values must be integers and must follow the DINT value range.

Example:

```
Card_game: ARRAY [1..13, 1..4] OF INT;
```

Initializing Arrays:**Example for complete initialization of an array:**

```
arr1 : ARRAY [1..5] OF INT := 1,2,3,4,5;
arr2 : ARRAY [1..2,3..4] OF INT := 1,3(7); (* short for 1,7,7,7 *)
arr3 : ARRAY [1..2,2..3,3..4] OF INT := 2(0),4(4),2,3;
      (* short for 0,0,4,4,4,4,2,3 *)
```

Example of the initialization of an array of a structure:

```
TYPE STRUCT1
STRUCT
p1:int;
p2:int;
p3:dword;
END_STRUCT
ARRAY[1..3] OF STRUCT1:= (p1:=1,p2:=10,p3:=4723), (p1:=2,p2:=0,p3:=299),
(p1:=14,p2:=5,p3:=112);
```

Example of the partial initialization of an Array:

```
arr1 : ARRAY [1..10] OF INT := 1,2;
```

Elements to which no value is pre-assigned are initialized with the default initial value of the basic type. In the example above, the elements anarray[6] to anarray[10] are therefore initialized with 0.

Accessing array components: v

Array components are accessed in a two-dimensional array using the following syntax:

```
<Field_Name>[Index1, Index2]
```

Example:

```
Card_game [9,2]
```

Note: If you define a function in your project with the name `CheckBounds`, you can use it to check for range overflows in your project (see chapter '2.1, What is what in CoDeSys', 'Components of a project', 'Function')

Function Checkbounds

If you define a function in your project with the name **CheckBounds**, you can automatically check for out-of-range errors in arrays. The name of the function is fixed and can only have this designation.

Example for the function CheckBounds:

```
FUNCTION CheckBounds : DINT
VAR_INPUT
  index, lower, upper: DINT;
END_VAR
IF index < lower THEN
  CheckBounds := lower;
ELSIF index > upper THEN
  CheckBounds := upper;
ELSE CheckBounds := index;
END_IF
```

The following sample program for testing the `CheckBounds` function exceeds the bounds of a defined array. The `CheckBounds` function allows the value `TRUE` to be assigned, not to location `A[10]`, but to the still valid range boundary `A[7]` above it. With the `CheckBounds` function, references outside of array boundaries can thus be corrected.

Test Program for the function CheckBounds:

```
PROGRAM PLC_PRG
VAR
  a: ARRAY[0..7] OF BOOL;
  b: INT:=10;
END_VAR
a[b]:=TRUE;
```

Attention: The `CheckBounds`-function provided by the `Check.Lib` library is just a sample solution! Before using that library module check whether the function is working as requested for your project, or implement an appropriate function directly as a POU in your project.

Pointer

Variable or function block addresses are saved in pointers while a program is running.

Pointer declarations have the following syntax:

```
<Identifier>: POINTER TO <Datatype/Functionblock>;
```

A pointer can point to any data type or function block even to user-defined types.

The function of the Address Operator `ADR` is to assign the address of a variable or function block to the pointer.

A pointer can be dereferenced by adding the content operator `^` after the pointer identifier.

Please note: A pointer is counted up byte-wise ! You can get it counted up like it is usual in the C-Compiler by using the instruction `p=p+SIZEOF(p^);`.

Example:

```
pt:POINTER TO INT;
var_int1:INT := 5;
var_int2:INT;
pt := ADR(var_int1);
var_int2:= pt^; (* var_int2 is now 5 *)
```

Functions for checking pointer accesses during runtime:

For checking pointer accesses during runtime you can create check functions, which will be called automatically before each access on the address of a pointer. For this purpose the respective function must be available in the project, directly or via a library. The following functions are supported:

Function **CheckPointer** or checking whether the address currently stored at the pointer is within the valid memory range,

Funktion **CheckPointerAligned**, which implicates the functionality of CheckPointer extended by a check of the memory alignment.

The functions must exactly have the mentioned names. They return the address which is used for dereferencing the pointer, thus at best that which has been passed on as the first input parameter (dwAddress in the example shown below).

See in the following example of a CheckPointerAligned function, which input parameters are processed by the check functions. (The parameter names are examples too). A CheckPointer function must look the same except that there may be no parameter for the granularity of the pointer access:

```

FUNCTION CheckPointerAligned : DWORD      (* The data type of the function (return value) must
                                          be the same as used for pointers
                                          in the currently set target system; i.e. DWORD for
                                          systems using 32-bit pointers,
                                          WORD for systems using 16-bit pointers *)

VAR_INPUT
  dwAddress : DWORD;                    (* Target address of the pointer; the data type must
                                          be the same as used for pointers
                                          in the currently set target system, see above:
                                          return value *)

  iSize : DINT;                          (* Size of pointer access; the data type must be
                                          integer-compatible and must cover the maximum
                                          potential data size stored at the pointer address *)

  iGran : DINT;                          (* !not to be used in CheckPointer functions!
                                          granularity of the access, e.g. "2", if INT is the
                                          smallest not-structured datatype
                                          used on the given address; the data type must be
                                          integer-compatible *)

  bWrite: BOOL;                          (*Access type: Read or Write; TRUE=read access;
                                          the data type must be BOOL *)


END_VAR

```

If there are a CheckPointer function and a CheckPointerAligned function in the project, CheckPointerAligned will be called.

Enumeration

Enumeration is a user-defined data type that is made up of a number of string constants. These constants are referred to as enumeration values.

Enumeration values are recognized in all areas of the project even if they were declared within a POU. It is best to create your enumerations as objects in the Object Organizer under the register card  **Data types**. They begin with the keyword TYPE and end with END_TYPE.

Syntax:

```

TYPE <Identifier>:(<Enum_0> ,<Enum_1>, ..., <Enum_n>);
END_TYPE

```

A variable of the type <Identifier> can take on one of the enumeration values and will be initialized with the first one. These values are compatible with whole numbers which means that you can perform operations with them just as you would with INT. You can assign a number x to the variable. If the enumeration values are not initialized, counting will begin with 0. When initializing, make certain the initial values are increasing. The validity of the number will be reviewed at the time it is run.

Example:

```

TYPE TRAFFIC SIGNAL: (Red, Yellow, Green:=10); (*The initial value for each of the
colors is red 0, yellow 1, green 10 *)
END_TYPE
TRAFFIC_SIGNAL1 : TRAFFIC_SIGNAL;
TRAFFIC_SIGNAL1:=0; (* The value of the traffic signal is red*)
FOR i:= Red TO Green DO
  i := i + 1;
END_FOR;

```

The same enumeration value may not be used twice within an enumeration or within all enumerations used in the same POU.

Example:

```

TRAFFIC_SIGNAL: (red, yellow, green);
COLOR: (blue, white, red);
Error: red may not be used for both TRAFFIC_SIGNAL and COLOR.

```

Structures

Structures are created as objects in the Object Organizer under the register card **Data types**. They begin with the keywords TYPE and STRUCT and end with END_STRUCT and END_TYPE.

The syntax for structure declarations is as follows:

```

TYPE <Structurename>:
STRUCT
  <Declaration of Variables 1>
  .
  .
  <Declaration of Variables n>
END_STRUCT
END_TYPE

```

<Structurename> is a type that is recognized throughout the project and can be used like a standard data type.

Interlocking structures are allowed. The only restriction is that variables may not be placed at addresses (the AT declaration is not allowed!).

Example for a structure definition named Polygonline:

```

TYPE Polygonline:
STRUCT
  Start:ARRAY [1..2] OF INT;
  Point1:ARRAY [1..2] OF INT;
  Point2:ARRAY [1..2] OF INT;
  Point3:ARRAY [1..2] OF INT;
  Point4:ARRAY [1..2] OF INT;
  End:ARRAY [1..2] OF INT;
END_STRUCT
END_TYPE

```

Example for the initialization of a structure:

```

Poly 1:polygonline := ( Start:=3,3, Point1 =5,2, Point2:=7,3, Point3:=8,5,
Point4:=5,7, End := 3,5);

```

Initializations with variables are not possible. See an example of the initialization of an array of a structure under 'Arrays'.

Access on structure components:

You can gain access to structure components using the following syntax:

```


<Structure_Name>.<Componentname>

```

So for the above mentioned example of the structure 'polygonline' you can access the component 'start' by `Poly_1.Start`.

References

You can use the user-defined reference data type to create an alternative name for a variable, constant or function block.

Create your references as objects in the Object Organizer under the register card  **Data types**. They begin with the keyword `TYPE` and end with `END_TYPE`.

Syntax:

```
TYPE <Identifier>: <Assignment term>;
END_TYPE
```

Example:

```
TYPE message:STRING[50];
END_TYPE;
```

Subrange types

A subrange type is a type whose range of values is only a subset of that of the basic type. The declaration can be carried out in the data types register, but a variable can also be directly declared with a subrange type:

Syntax for the declaration in the 'Data types' register:

```
TYPE <Name> : <Inttype> (<ug>..<og>) END_TYPE;
```

<Name> must be a valid IEC identifier,

<Inttype> is one of the data types SINT, USINT, INT, UINT, DINT, UDINT, BYTE, WORD, DWORD (LINT, ULINT, LWORD).

<ug> Is a constant which must be compatible with the basic type and which sets the lower boundary of the range types. The lower boundary itself is included in this range.

<og> Is a constant that must be compatible with the basic type, and sets the upper boundary of the range types. The upper boundary itself is included in this basic type.

Examples:

```
TYPE
  SubInt : INT (-4095..4095);
END_TYPE
```

Direct declaration of a variable with a subrange type:

```
VAR
  i : INT (-4095..4095);
  ui : UINT (0..10000);
END_VAR
```

If a constant is assigned to a subrange type (in the declaration or in the implementation) that does not fall into this range (e.g. `1:=5000`), an error message is issued.

In order to check for observance of range boundaries at runtime, the functions **CheckRangeSigned** or **CheckRangeUnsigned** must be introduced. In these, boundary violations can be captured by the appropriate method and means (e.g. the value can be cut out or an error flag can be set.). They are implicitly called as soon as a variable is written as belonging to a subrange type constructed from either a signed or an unsigned type.

Example:

In the case of a variable belonging to a signed subrange type (like `i`, above), the function `CheckRangeSigned` is called; it could be programmed as follows to trim a value to the permissible range:

```
FUNCTION CheckRangeSigned : DINT
VAR_INPUT
  value, lower, upper: DINT;
```

```

END_VAR
IF (value < lower) THEN
  CheckRangeSigned := lower;
ELSIF(value > upper) THEN
  CheckRangeSigned := upper;
ELSE
  CheckRangeSigned := value;
END_IF

```

In calling up the function automatically, the function name CheckRangeSigned is obligatory, as is the interface specification: return value and three parameters of type DINT

When called, the function is parameterized as follows:

- value: the value to be assigned to the range type
- lower: the lower boundary of the range
- upper: the upper boundary of the range
- Return value: this is the value that is actually assigned to the range type

An assignment `i:=10*y` implicitly produces the following in this example:

```
i := CheckRangeSigned(10*y, -4095, 4095);
```

Even if `y` for example has the value 1000, then `i` still has only the value 4095 after this assignment.

The same applies to the function CheckRangeUnsigned: function name and interface must be correct.

```

FUNCTION CheckRangeUnsigned : UDINT
VAR_INPUT
  value, lower, upper: UDINT;
END_VAR

```

Important: If neither of the functions CheckRangeSigned or CheckRangeUnsigned is present, no type checking of subrange types occurs during runtime! The variable `i` could then take on any value between -32768 and 32767 at any time!

Attention: If neither of the functions CheckRangeSigned or CheckRangeUnsigned is present like described above, there can result an endless loop if a subrange type is used in a **FOR** loop. This will happen when the range given for the FOR loop is as big or bigger than the range of the subrange type !

Attention: The CheckRangeSigned-function provided with the Check.Lib library is just a sample solution! Before using the library module check whether the function is working as requested for your project, or implement an appropriate CheckRange-function directly as a POU in the project.

Example:

```

VAR
  ui : UINT (0..10000);
END_VAR

FOR ui:=0 TO 10000 DO
  ...
END_FOR

```

The FOR loop will never be finished, because `ui` cannot get bigger than 10000.

Also take care of the definition of the CheckRange functions when you define the incremental value of a FOR loop !

Appendix D: The CoDeSys Libraries

10.17 The Standard.lib library

10.17.1 String functions...

Please note: String functions are not "thread safe": When using tasks, string functions may only be used in a single task. If the same function is used in different tasks, there is a danger of overwriting.

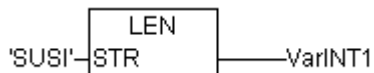
LEN

Returns the length of a string. Input STR is of type STRING, the return value of the function is type INT.

Example in IL:

```
LD 'SUSI'
LEN
ST VarINT1 (* Result is 4 *)
```

Example in FBD:



Example in ST:

```
VarSTRING1 := LEN ('SUSI');
```

LEFT

Left returns the left, initial string for a given string. Input STR is type STRING, SIZE is of type INT, the return value of the function is type STRING.

LEFT (STR, SIZE) means: Take the first SIZE character from the right in the string STR.

Example in IL:

```
LD 'SUSI'
LEFT 3
ST VarSTRING1 (* Result is 'SUS' *)
```

Example in FBD:



Example in ST:

```
VarSTRING1 := LEFT ('SUSI',3);
```

RIGHT

Right returns the right, initial string for a given string.

RIGHT (STR, SIZE) means: Take the first SIZE character from the right in the string STR.

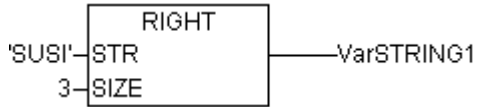
Input STR is of type STRING, SIZE is of type INT, the return value of the function is of type STRING.

Example in IL:

```
LD 'SUSI'
```

```
RIGHT 3
ST VarSTRING1 (* Result is 'USI' *)
```

Example in FBD:



Example in ST:

```
VarSTRING1 := RIGHT ('SUSI',3);
```

MID

Mid returns a partial string from within a string.

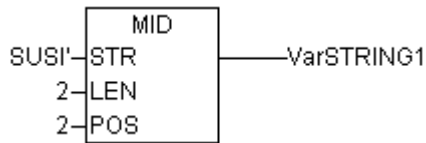
Input STR is type STRING, LEN and POS are type INT, the return value of the function is type STRING.

MID (STR, LEN, POS) means: Retrieve LEN characters from the STR string beginning with the character at position POS.

Example in IL:

```
LD 'SUSI'
MID 2,2
ST VarSTRING1 (* Result is 'US' *)
```

Example in FBD:



Example in ST:

```
VarSTRING1 := MID ('SUSI',2,2);
```

CONCAT

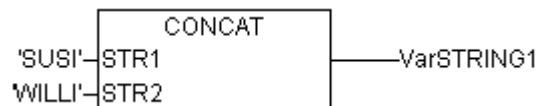
Concatenation (combination) of two strings.

The input variables STR1 and STR2 as well as the return value of the function are type STRING.

Example in IL:

```
LD 'SUSI'
CONCAT 'WILLI'
ST VarSTRING1 (* Result is 'SUSIWILLI' *)
```

Example in FBD:



Example in ST:

```
VarSTRING1 := CONCAT ('SUSI','WILLI');
```

Please note: The CONCAT function does not work, if nested over more than five levels.

INSERT

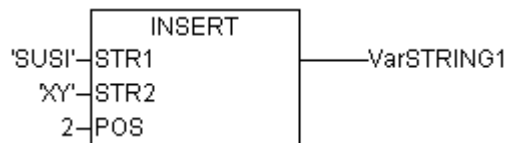
INSERT inserts a string into another string at a defined point.

The input variables STR1 and STR2 are type STRING, POS is type INT and the return value of the function is type STRING.

INSERT(STR1, STR2, POS) means: insert STR2 into STR1 after position POS.

Example in IL:

```
LD    'SUSI'
INSERT 'XY',2
ST    VarSTRING1 (* Result is 'SUXYSI' *)
```

Example in FBD:**Example in ST:**

```
VarSTRING1 := INSERT ('SUSI','XY',2);
```

DELETE

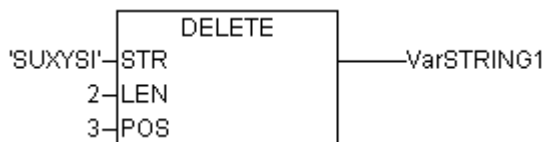
DELETE removes a partial string from a larger string at a defined position.

The input variable STR is type STRING, LEN and POS are type INT, the return value of the function is type STRING.

DELETE(STR, L, P) means: Delete L characters from STR beginning with the character in the P position.

Example in IL:

```
LD    'SUXYSI'
DELETE 2,3
ST    Var1 (* Result is 'SUSI' *)
```

Example in FBD:**Example in ST:**

```
Var1 := DELETE ('SUXYSI',2,3);
```

REPLACE

REPLACE replaces a partial string from a larger string with a third string.

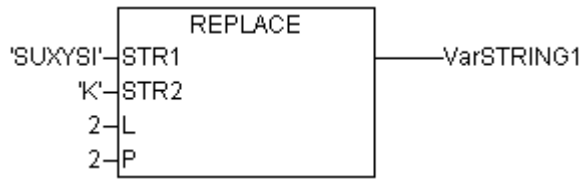
The input variable STR1 and STR2 are type STRING, LEN and POS are type INT, the return value of the function is type STRING.

REPLACE(STR1, STR2, L, P) means: Replace L characters from STR1 with STR2 beginning with the character in the P position.

Example in IL:

```
LD    'SUXYSI'
REPLACE 'K',2,2
ST    VarSTRING1 (* Result is 'SKYSI' *)
```

Example in FBD:



Example in ST:

```
VarSTRING1 := REPLACE ('SUXYSI','K',2,2);
```

FIND

FIND searches for a partial string within a string.

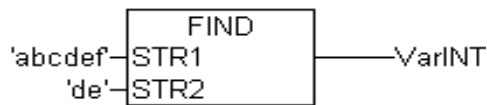
The input variable STR1 and STR2 are type STRING, the return value of the function is type STRING.

FIND(STR1, STR2) means: Find the position of the first character where STR2 appears in STR1 for the first time. If STR2 is not found in STR1, then OUT:=0.

Example in IL:

```
LD 'abcdef'
FIND 'de'
ST VarINT1 (* Result is '4' *)
```

Example in FBD:



Example in ST:

```
VarINT1 := FIND ('abcdef','de');
```

10.17.2 Bistable Function Blocks...

SR

Making Bistable Function Blocks Dominant:

Q1 = SR (SET1, RESET) means:

$$Q1 = (\text{NOT RESET AND } Q1) \text{ OR SET1}$$

The input variables SET1 and RESET as well as the output variable Q1 are type BOOL.

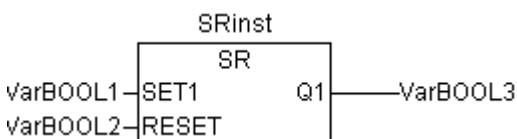
Declaration example:

```
SRInst : SR ;
```

Example in IL:

```
CAL SRInst(SET1 := VarBOOL1, RESET := VarBOOL2)
LD SRInst.Q1
ST VarBOOL3
```

Example in FBD:



Example in ST:

```
SRInst(SET1:= VarBOOL1 , RESET:=VarBOOL2 );
VarBOOL3 := SRInst.Q1 ;
```

RS**Resetting Bistable Function Blocks**

Q1 = RS (SET, RESET1) means:

$$Q1 = \text{NOT RESET1 AND } (Q1 \text{ OR SET})$$

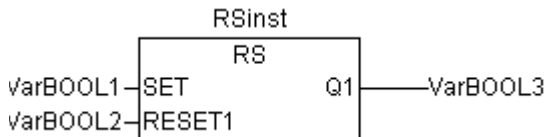
The input variables SET and RESET1 as well as the output variable Q1 are type BOOL.

Declaration example:

```
RSInst : RS ;
```

Example in IL:

```
CAL RSInst(SET:= VarBOOL1,RESET1:=VarBOOL2)
LD RSInst.Q1
ST VarBOOL3
```

Example in FBD:**Example in ST:**

```
RSInst(SET:= VarBOOL1 , RESET1:=VarBOOL2 );
VarBOOL3 := RSInst.Q1 ;
```

SEMA**A Software Semaphore (Interruptible)**

BUSY = SEMA(CLAIM, RELEASE) means:

```
BUSY := X;
IF CLAIM THEN X:=TRUE;
ELSE IF RELEASE THEN BUSY := FALSE; X:= FALSE;
END_IF
```

X is an internal BOOL variable that is FALSE when it is initialized. The input variables CLAIM and RELEASE as well as the output variable BUSY are type BOOL.

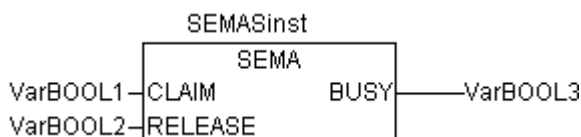
If BUSY is TRUE when SEMA is called up, this means that a value has already been assigned to SEMA (SEMA was called up with CLAIM = TRUE). If BUSY is FALSE, SEMA has not yet been called up or it has been released (called up with RELEASE = TRUE).

Declaration example:

```
SEMAInst : SEMA ;
```

Example in IL:

```
CAL SEMAInst(CLAIM:=VarBOOL1,RELEASE:=VarBOOL2)
LD SEMAInst.BUSY
ST VarBOOL3
```

Example in FBD:

Example in ST:

```
SEMAInst (CLAIM:= VarBOOL1 , RELEASE:=VarBOOL2 );
VarBOOL3 := SEMAInst.BUSY;
```

10.17.3 Trigger...**R_TRIG**

The function block R_TRIG detects a rising edge.

```
FUNCTION_BLOCK R_TRIG
VAR_INPUT
  CLK : BOOL;
END_VAR
VAR_OUTPUT
  Q : BOOL;
END_VAR
VAR
  M : BOOL := FALSE;
END_VAR
  Q := CLK AND NOT M;
  M := CLK;
```

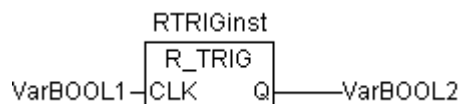
The output Q and the help variable M will remain FALSE as long as the input variable CLK is FALSE. As soon as CLK returns TRUE, Q will first return TRUE, then M will be set to TRUE. This means each time the function is called up, Q will return FALSE until CLK has falling edge followed by an rising edge.

Declaration example:

```
RTRIGInst : R_TRIG ;
```

Example in IL:

```
CAL RTRIGInst (CLK := VarBOOL1)
LD RTRIGInst.Q
ST VarBOOL2
```

Example in FBD:**Example in ST:**

```
RTRIGInst (CLK:= VarBOOL1);
VarBOOL2 := RTRIGInst.Q;
```

F_TRIG

The function block F_TRIG a falling edge.

```
FUNCTION_BLOCK F_TRIG
VAR_INPUT
  CLK: BOOL;
END_VAR
VAR_OUTPUT
  Q: BOOL;
END_VAR
VAR
END_VAR
```

```

M: BOOL := FALSE;
END_VAR
Q := NOT CLK AND NOT M;
M := NOT CLK;

```

The output Q and the help variable M will remain FALSE as long as the input variable CLK returns TRUE. As soon as CLK returns FALSE, Q will first return TRUE, then M will be set to TRUE. This means each time the function is called up, Q will return FALSE until CLK has a rising followed by a falling edge.

Declaration example:

```
FTRIGInst : F_TRIG ;
```

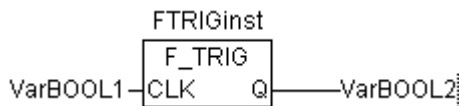
Example in IL:

```

CAL FTRIGInst(CLK := VarBOOL1)
LD FTRIGInst.Q
ST VarBOOL2

```

Example in FBD:



Example in ST:

```

FTRIGInst(CLK:= VarBOOL1);
VarBOOL2 := FTRIGInst.Q;

```

10.17.4 Counter...

CTU

Function block Incrementer:

The input variables CU and RESET as well as the output variable Q are type BOOL, the input variable PV and the output variable CV are type WORD.

The counter variable CV will be initialized with 0 if RESET is TRUE. If CU has a rising edge from FALSE to TRUE, CV will be raised by 1. Q will return TRUE when CV is greater than or equal to the upper limit PV.

Declaration example:

```
CTUInst : CTU ;
```

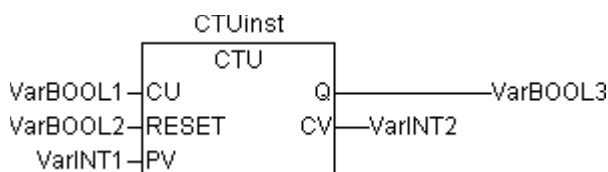
Example in IL:

```

CAL CTUInst(CU := VarBOOL1, RESET := VarBOOL2, PV := VarINT1)
LD CTUInst.Q
ST VarBOOL3
LD CTUInst.CV
ST VarINT2

```

Example in FBD:



Example in ST:

```
CTUInst(CU:= VarBOOL1, RESET:=VarBOOL2 , PV:= VarINT1);
VarBOOL3 := CTUInst.Q ;
VarINT2 := CTUInst.CV;
```

CTD

Function Block Decrementer:

The input variables CD and LOAD as well as the output variable Q are type BOOL, the input variable PV and the output variable CV are type WORD.

When LOAD_ is TRUE, the counter variable CV will be initialized with the upper limit PV. If CD has a rising edge from FALSE to TRUE, CV will be lowered by 1 provided CV is greater than 0 (i.e., it doesn't cause the value to fall below 0).

Q returns TRUE when CV is equal 0.

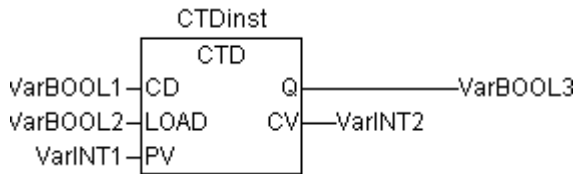
Declaration example:

```
CTDInst : CTD ;
```

Example in IL:

```
CAL CTDInst(CD := VarBOOL1, LOAD := VarBOOL2, PV := VarINT1)
LD CTDInst.Q
ST VarBOOL3
LD CTDInst.CV
ST VarINT2
```

Example in FBD:



Example in ST:

```
CTDInst(CD:= VarBOOL1, LOAD:=VarBOOL2 , PV:= VarINT1);
VarBOOL3 := CTDInst.Q ;
VarINT2 := CTDInst.CV;
```

CTUD

Function Block Incrementer/Decrementer

The input variables CU, CD, RESET, LOAD as well as the output variables QU and QD are type BOOL, PV and CV are type WORD.

If RESET is valid, the counter variable CV will be initialized with 0. If LOAD is valid, CV will be initialized with PV.

If CU has a rising edge from FALSE to TRUE, CV will be raised by 1. If CD has a rising edge from FALSE to TRUE, CV will be lowered by 1 provided this does not cause the value to fall below 0.

QU returns TRUE when CV has become greater than or equal to PV.

QD returns TRUE when CV has become equal to 0.

Declaration example:

```
CTUDInst : CUTD ;
```

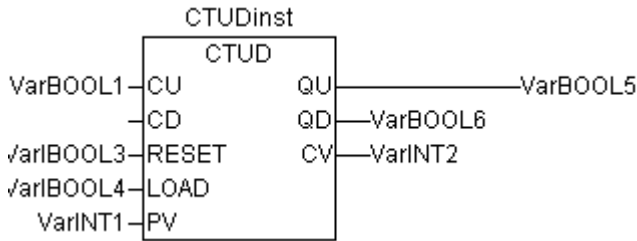
Example in IL:

```
CAL CTUDInst(CU:=VarBOOL2, RESET:=VarBOOL3, LOAD:=VarBOOL4, PV:=VarINT1)
LD CTUDInst.Q
```

```

ST   VarBOOL5
LD   CTUDInst.QD
ST   VarBOOL5
LD   CTUInst.CV
ST   VarINT2
    
```

Example in FBD:



Example in ST:

```

CTUDInst(CU := VarBOOL1, CU:= VarBOOL2, RESET := VarBOOL3, LOAD:=VarBOOL4 , PV:=
VarINT1);
VarBOOL5 := CTUDInst.QU ;
VarBOOL6 := CTUDInst.QD ;
VarINT2 := CTUDInst.CV;
    
```

10.17.5 Timer...

TP

The function block Timer is a trigger. TP(IN, PT, Q, ET) means:

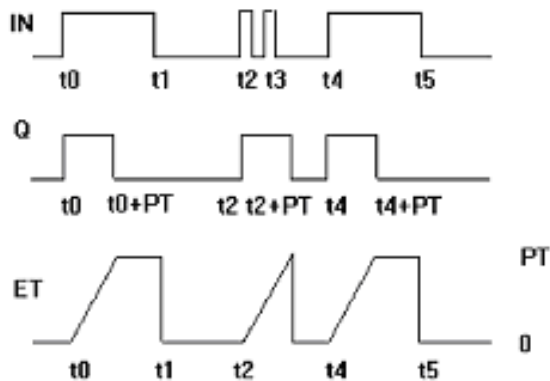
IN and PT are input variables of the BOOL and TIME types respectively. Q and ET are output variables of the BOOL and TIME types respectively. If IN is FALSE, Q is FALSE and ET is 0.

As soon as IN becomes TRUE, the time will begin to be counted in milliseconds in ET until its value is equal to PT. It will then remain constant.

Q is TRUE as from IN has got TRUE and ET is less than or equal to PT. Otherwise it is FALSE.

Q returns a signal for the time period given in PT.

Graphic Display of the TP Time Sequence



Declaration example:

```

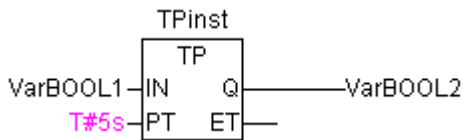
TPInst : TP ;
    
```

Example in IL:

```

CAL TPInst(IN := VarBOOL1, PT := T#5s)
LD   TPInst.Q
ST   VarBOOL2
    
```

Example in FBD:



Example in ST:

```
TPInst(IN := VarBOOL1, PT:= T#5s);
VarBOOL2 :=TPInst.Q;
```

TON

The function block Timer On Delay implements a turn-on delay..

TON(IN, PT, Q, ET) means:

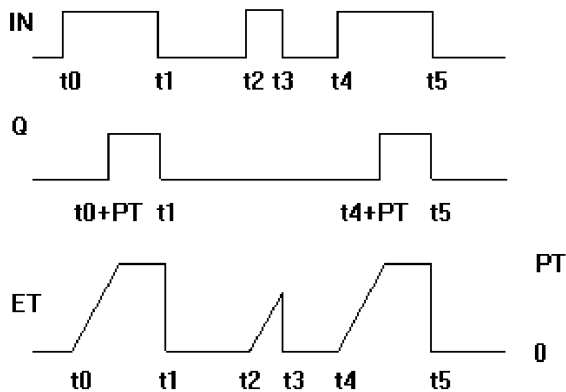
IN and PT are input variables of the BOOL and TIME types respectively. Q and ET are output variables of the BOOL and TIME types respectively. If IN is FALSE, Q is FALSE and ET is 0.

As soon as IN becomes TRUE, the time will begin to be counted in milliseconds in ET until its value is equal to PT. It will then remain constant.

Q is TRUE when IN is TRUE and ET is equal to PT. Otherwise it is FALSE.

Thus, Q has a rising edge when the time indicated in PT in milliseconds has run out.

Graphic display of TON behaviour over time:



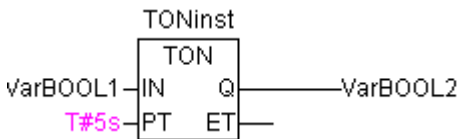
Declaration example:

```
TONInst : TON ;
```

Example in IL:

```
CAL TONInst(IN := VarBOOL1, PT := T#5s)
LD TONInst.Q
ST VarBOOL2
```

Example in FBD:



Example in ST:

```
TONInst(IN := VarBOOL1, PT:= T#5s);
```

TOF

The function block TOF implements a turn-off delay..

TOF(IN, PT, Q, ET) means:

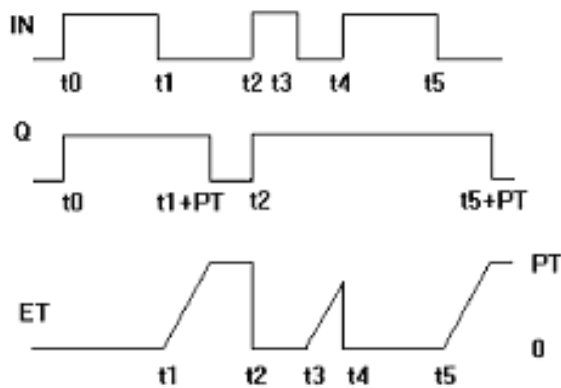
IN and PT are input variables type BOOL respectively TIME. Q and E are output variables type BOOL respectively TIME. If IN is TRUE, the outputs are TRUE respectively 0.

As soon as IN becomes FALSE, in ET the time will begin to be counted in milliseconds in ET until its value is equal to PT. It will then remain constant.

Q is FALSE when IN is FALSE und ET equal PT. Otherwise it is TRUE.

Thus, Q has a falling edge when the time indicated in PT in milliseconds has run out.

Graphic display of TOF behaviour over time:



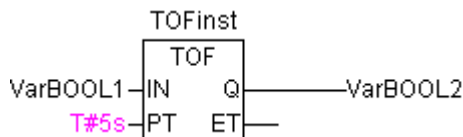
Declaration example:

```
TOFInst : TOF ;
```

Example in IL:

```
CAL TOFInst(IN := VarBOOL1, PT := T#5s)
LD TOFInst.Q
ST VarBOOL2
```

Example in FBD:

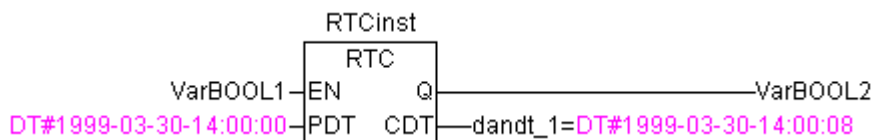


Example in ST:

```
TOFInst(IN := VarBOOL1, PT:= T#5s);
VarBOOL2 :=TOFInst.Q;
```

RTC

The function block Runtime Clock returns, starting at a given time, the current date and time.



RTC(EN, PDT, Q, CDT) means:

EN and PDT are input variables type TIME. Q and CDT are output variables type BOOL respectively DATE_AND_TIME. When EN is FALSE, the output variables Q and CDT are FALSE respectively DT#1970-01-01-00:00:00.

As soon as EN becomes TRUE, the time of PDT is set, is counted up in seconds and returned in CDT as long as EN is TRUE (see example in the picture above). As soon as EN is reset to FALSE, CDT is reset to the initial value DT#1970-01-01-00:00:00. Please note that the time in PDT is only set by a rising edge.

10.18 The Util.lib library

This library contains an additional collection of various blocks which can be used for BCD conversion, bit/byte functions, mathematical auxiliary functions, as controller, signal generators, function manipulators and for analogue value processing.

As some of the functions and function blocks contain REAL variables, an accessory library named UTIL_NO_REAL exists in which these POU's are excluded.

10.18.1 BCD Conversion

A byte in the BCD format contains integers between 0 and 99. Four bits are used for each decimal place. The ten decimal place is stored in the bits 4-7. Thus the BCD format is similar to the hexadecimal presentation, with the simple difference that only values between 0 and 99 can be stored in a BCD byte, whereas a hexadecimal byte reaches from 0 to FF.

An example: The integer 51 should be converted to BCD format. 5 in binary is 0101, 1 in binary is 0001, which makes the BCD byte 01010001, which corresponds to the value \$51=81.

BCD_TO_INT

This function converts a byte in BCD format into an INT value:

The input value of the function is type BYTE and the output is type INT.

Where a byte should be converted which is not in the BCD format the output is -1.

Examples in ST:

```
i:=BCD_TO_INT(73); (* Result is 49 *)
k:=BCD_TO_INT(151); (* Result is 97 *)
l:=BCD_TO_INT(15); (* Output -1, because it is not in BCD format *)
```

INT_TO_BCD

This function converts an INTEGER value into a byte in BCD format:

The input value of the function is type INT, the output is type BYTE.

The number 255 will be outputted where an INTEGER value should be converted which cannot be converted into a BCD byte.

Examples in ST:

```
i:=INT_TO_BCD(49); (* Result is 73 *)
k:=INT_TO_BCD(97); (* Result is 151 *)
l:=INT_TO_BCD(100); (* Error! Output: 255 *)
```

10.18.2 Bit-/Byte Functions

EXTRACT

Inputs to this function are a DWORD X, as well as a BYTE N. The output is a BOOL value, which contains the content of the Nth bit of the input X, whereby the function begins to count from the zero bit.

Examples in ST:

```
FLAG:=EXTRACT(X:=81, N:=4); (* Result : TRUE, because 81 is binary 1010001, so the
4th bit is 1 *)
FLAG:=EXTRACT(X:=33, N:=0); (* Result : TRUE, because 33 is binary 100001, so the
bit '0' is 1 *)
```

PACK

This function is capable of delivering back eight input bits B0, B1, ..., B7 from type BOOL as a BYTE. The function block UNPACK is closely related to this function.

PUTBIT

The input to this function consists of a DWORD X, a BYTE N and a BOOLEAN value B. PUTBIT sets the Nth bit from X on the value B, whereby it starts counting from the zero bit.

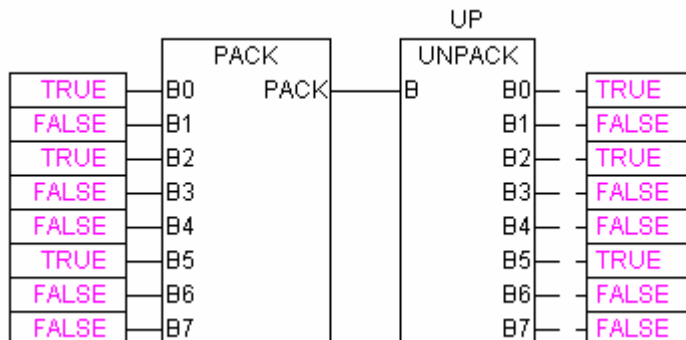
Example in ST:

```
A:=38;      (* binary 100110 *)
B:=PUTBIT(A,4,TRUE); (* Result : 54 = 2#110110 *)
C:=PUTBIT(A,1,FALSE); (* Result : 36 = 2#100100 *)
```

UNPACK

UNPACK converts the input B from type BYTE into 8 output variables B0, ..., B7 of the type BOOL, and this is the opposite to PACK.

Example in FBD: Output:

**10.18.3 Mathematic Auxiliary Functions****DERIVATIVE**

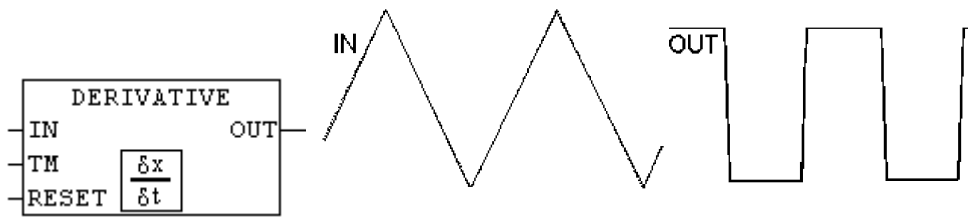
This function block approximately determines the local derivation.

The function value is delivered as a REAL variable by using IN. TM contains the time which has passed in msec in a DWORD and the input of RESET of the type BOOL allows the function block to start anew through the delivery of the value TRUE.

The output OUT is of the type REAL.

In order to obtain the best possible result, DERIVATIVE approximates using the last four values, in order to hold errors which are produced by inaccuracies in the input parameters as low as possible.

Block in FBD:



INTEGRAL

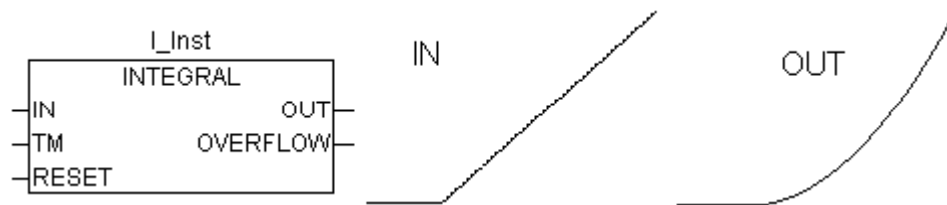
This function block approximately determines the integral of the function.

In an analogue fashion to DERIVATIVE, the function value is delivered as a REAL variable by using IN. TM contains the time which has passed in msec in a DWORD and the input of RESET of the type BOOL allows the function block to start anew with the value TRUE.

The output OUT is of the type REAL.

The integral is approximated by two step functions. The average of these is delivered as the approximated integral.

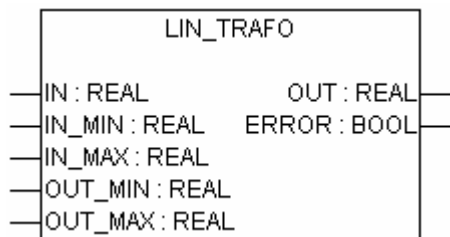
Block in FBD: Example: Integration of a linear function:



LIN_TRAFO

This function block (util.lib. transforms a REAL-value, which lies in a value range defined by a lower and upper limit value, to a corresponding REAL-value which lies in another range also defined by a lower and upper limit. The following equation is basis of the conversion:

$$(IN - IN_MIN) : (IN_MAX - IN) = (OUT - OUT_MIN) : (OUT_MAX - OUT)$$



Input variables:

Variable	Data type	Description
IN	REAL	Input value
IN_MIN	REAL	Lower limit of input value range
IN_MAX	REAL	Upper limit of input value range
OUT_MIN	REAL	Lower limit of output value range
OUT_MAX	REAL	Upper limit of output value range

Output variables:

Variable	Data type	Description
OUT	REAL	Output value
ERROR	BOOL	Error occurred: TRUE, if IN_MIN = IN_MAX, or if IN is outside of the specified input value range

Application example:

A temperature sensor provides Volt-values (input IN). These are to be converted to temperature values in degree centigrade (output OUT). The input(Volt) values range is defined by the limits IN_MIN=0 and IN_MAX=10. The output(degree centigrade) value range is defined by the limits OUT_MIN=-20 and OUT_MAX=40.

Thus for an input of 5 Volt a temperature of 10 degree centigrade will result.

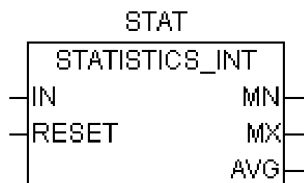
STATISTICS_INT

This function block calculates some standard statistical values:

The input IN is of the type INT. All values are initialised anew when the Boolean input RESET is TRUE.

The output MN contains the minimum, MX of the maximum value from IN. AVG describes the average, that is the expected value of IN. All three outputs are of the type INT.

Block in FBD:



STATISTICS_REAL

This function block corresponds to STATISTICS_INT, except that the input IN is of the type REAL like the outputs MN, MX, AVG.

VARIANCE

VARIANCE calculates the variance of the entered values.

The input IN is of the type REAL, RESET is of the type BOOL and the output OUT is again of the type REAL.

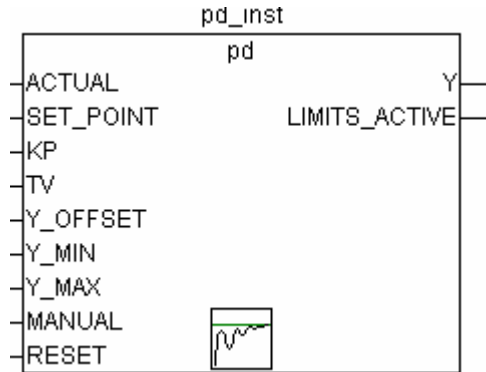
This block calculates the variance of the inputted values. VARIANCE can be reset with RESET=TRUE.

The standard deviation can easily be calculated as the square root of the VARIANCE.

10.18.4 Controllers

PD

The library util.lib provides the following PD controller function block:



Inputs of the function block:

Variable	Data type	Description
ACTUAL	REAL	Current value of the controlled variable
SET_POINT	REAL	Desired value, command variable
KP	REAL	Proportionality coefficient, unity gain of the P-part
TV	REAL	Derivative action time, unity gain of the D-part in seconds, e.g. "0.5" for 500 msec
Y_MANUAL	REAL	Defines output value Y in case of MANUAL = TRUE
Y_OFFSET	REAL	Offset for the manipulated variable Y
Y_MIN, Y_MAX	REAL	Lower resp. upper limit for the manipulated variable Y. If Y exceeds these limits, output LIMITS_ACTIVE will be set to TRUE and Y will be kept within the prescribed range. This control will only work if Y_MIN < Y_MAX.
MANUAL	BOOL	If TRUE, manual operation will be active, i.e. the manipulated value will be defined by Y_MANUAL.
RESET	BOOL	TRUE resets the controller; during reinitialization Y = Y_OFFSET.

Outputs of the function block:

Variable	Data type	Description
Y	REAL	Manipulated value, calculated by the function block (see below)
LIMITS_ACTIVE	BOOL	TRUE indicates that Y has exceeded the given limits (Y_MIN, Y_MAX).

Y_OFFSET, Y_MIN and Y_MAX are used for the transformation of the manipulated variable within a prescribed range.

MANUAL can be used to switch on and off manual operation. RESET serves to reset the controller.

In normal operation (MANUAL = RESET = LIMITS_ACTIVE = FALSE) the controller calculates the **controller error e** as difference SET_POINT – ACTUAL, generates the derivation with respect to time $\delta e / \delta t$ and stores these values internally.

The output, i.e. the **manipulated variable Y** is calculated as follows:

$$Y = KP \cdot (\Delta + TV \delta\Delta/\delta t) + Y_OFFSET \text{ whereby } \Delta = \text{SET_POINT} - \text{ACTUAL}$$

So besides the P-part also the current change of the controller error (D-part) influences the manipulated variable.

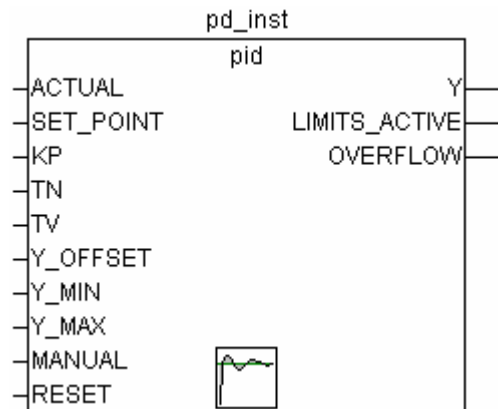
Additionally Y is limited on a range prescribed by Y_MIN and Y_MAX. If Y exceeds these limits, LIMITS_ACTIVE will get TRUE. If no limitation of the manipulated variable is desired, Y_MIN and Y_MAX have to be set to 0.

As long as MANUAL=TRUE, Y_MANUAL will be written to Y.

A **P-controller** can be easily created by setting TV=0.

PID

The library util.lib provides the following PID controller function block:



Unlike the PD controller, this function block contains a further REAL input TN for the readjusting time in sec (e.g. "0.5" for 500 msec).

Inputs of the function block:

Variable	Data type	Description
ACTUAL	REAL	Current value of the controlled variable
SET_POINT	REAL	Desired value, command variable
KP	REAL	Proportionality coefficient, unity gain of the P-part
TN	REAL	Reset time, reciprocal unity gain of the I-part; given in seconds, e.g. "0.5" for 500 msec
TV	REAL	Derivative action time, unity gain of the D-part in seconds, e.g. "0.5" for 500 msec
Y_MANUAL	REAL	Defines output value Y in case of MANUAL = TRUE
Y_OFFSET	REAL	Offset for the manipulated variable Y
Y_MIN, Y_MAX	REAL	Lower resp. upper limit for the manipulated variable Y. If Y exceeds these limits, output LIMITS_ACTIVE will be set to TRUE and Y will be kept within the prescribed range. This control will only work if Y_MIN < Y_MAX.
MANUAL	BOOL	If TRUE, manual operation will be active, i.e. the manipulated

		value will be defined by Y_MANUAL.
RESET	BOOL	TRUE resets the controller; during reinitialization $Y = Y_OFFSET$.

Outputs of the function block:

Variable	Data type	Description
Y	REAL	Manipulated value, calculated by the function block (see below)
LIMITS_ACTIVE	BOOL	TRUE indicates that Y has exceeded the given limits (Y_MIN, Y_MAX).
OVERFLOW	BOOL	TRUE indicates an overflow (see below)

Y_OFFSET, Y_MIN und Y_MAX serve for transformation of the manipulated variable within a prescribed range.

MANUAL can be used to switch to manual operation; RESET can be used to re-initialize the controller..

In normal operation (MANUAL = RESET = LIMITS_ACTIVE = FALSE) the controller calculates the **controller error e** as difference from SET_POINT – ACTUAL, generates the derivation with respect to time $\delta e/\delta t$ and stores these values internally.

The output, i.e. the manipulated variable Y unlike the PD controller contains an additional integral part and is calculated as follows:

$$Y = KP \cdot (\Delta + 1/TN \int e dt + TV \delta \Delta/\delta t) + Y_OFFSET$$

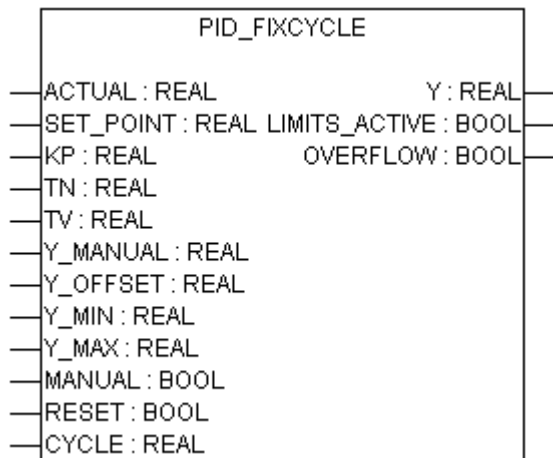
So besides the **P-part** also the current change of the controller error (**D-part**) and the history of the controller error (**I-part**) influence the manipulated variable.

The PID controller can be easily converted to a **PI-controller** by setting TV=0.

Because of the additional integral part, an **overflow** can come about by incorrect parameterization of the controller, if the integral of the error Δ becomes to great. Therefore for the sake of safety a Boolean output called OVERFLOW is present, which in this case would have the value TRUE. This only will happen if the control system is instable due to incorrect parameterization. At the same time, the controller will be suspended and will only be activated again by re-initialization.

PID_FIXCYCLE

The library util.lib provides the following PID_FIXCYCLE controller function block:



This function block functionally corresponds to the PID controller with the exception that the cycle time is not measured automatically by an internal function but is set by input CYCLE (in seconds).

10.18.5 Signal Generators...

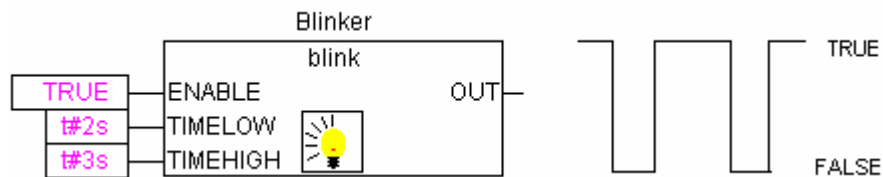
BLINK

The function block BLINK generates a pulsating signal. The input consists of ENABLE of the type BOOL, as well as TIMELOW and TIMEHIGH of the type TIME. The output OUT is of the type BOOL.

If ENABLE is set to TRUE, BLINK begins, to set the output for the time period TIMEHIGH to TRUE, and then afterwards to set it for the time period TIMELOW to FALSE.

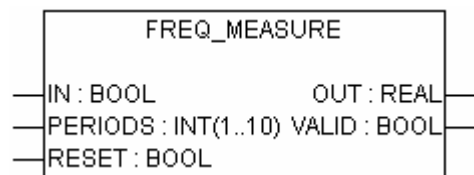
When ENABLE is reset to FALSE, output OUT will not be changed, i.e. no further pulse will be generated. If you explicitly also want to get OUT FALSE when ENABLE is reset to FALSE, you might use "OUT AND ENABLE" (i.e. adding an AND box with parameter ENABLE) at the output.

Example in CFC:



FREQ_MEASURE

This function block measures the (average) frequency (Hz) of a boolean input signal. You can specify over how many periods it should be averaged. A period is the time between two rising edges of the input signal.



Input variables:

Variable	Data type	Description
IN	BOOL	Input signal
PERIODS	INT	Number of periods, i.e. the time intervals between the rising edges, over which the average frequency of the input signal should be calculated. Possible values: 1 to 10
RESET	BOOL	Reset of all parameters to 0

Output variables:

Variable	Data type	Description
OUT	REAL	resulting frequency in [Hz]
VALID	BOOL	FALSE until the first measure has been finished, or if the period > 3*OUT (indicating something wrong with the inputs)

GEN

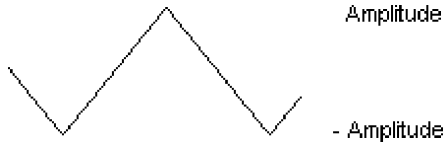
The function generator generates typical periodic functions:

The inputs are a composition consisting of MODE from the pre-defined counting type GEN_MODE, BASE of the type BOOL, PERIOD of the type TIME, of two INT values CYCLES and AMPLITUDE and of the Boolean RESET input.

The MODE describes the function which should be generated, whereby the enumeration values TRIANGLE and TRIANGLE_POS deliver two triangular functions, SAWTOOTH_RISE an ascending,

SAWTOOTH_FALL a descending sawtooth, RECTANGLE a rectangular signal and SINE and COSINE the sine and cosine:

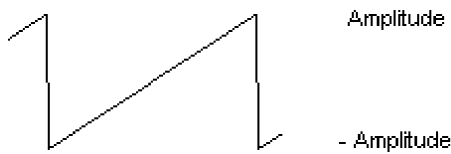
TRIANGLE:



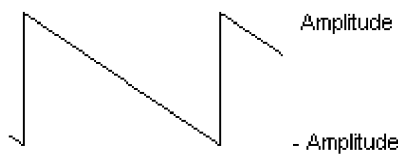
TRIANGLE_POS:



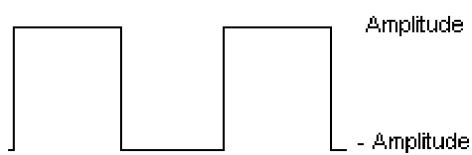
SAWTOOTH_RISE:



SAWTOOTH_FALL:



RECTANGLE:



SINUS:



COSINUS:



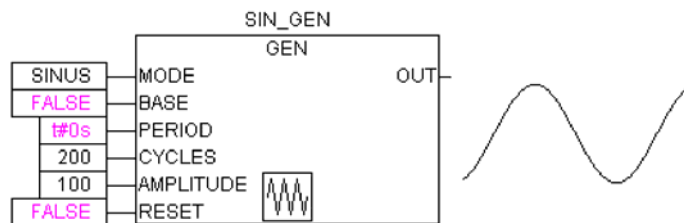
BASE defines whether the cycle period is really related to a defined time (BASE=TRUE) or whether it is related to a particular number of cycles, which means the number of calls of function block (BASE=FALSE).

PERIOD or CYCLES defines the corresponding cycle period.

AMPLITUDE defines, in a trivial way, the amplitude of the function to be generated.

The function generator is again set to 0 as soon as RESET=TRUE.

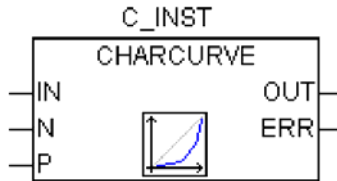
Example in FBD:



10.18.6 Function Manipulators...

CHARCURVE

This function block serves to represent values, piece by piece, on a linear function:



IN of the type INT is fed with the value to be manipulated. The BYTE N designates the number of points which defines the presentation function. This characteristic line is then generated in an ARRAY P[0..10] with P of the type POINT which is a structure based on two INT values (X and Y).

The output consists of OUT of the type INT, the manipulated value and BYTE ERR, which will indicate an error if necessary.

The points P[0]..P[N-1] in the ARRAY must be sorted according to their X values, otherwise ERR receives the value 1. If the input IN is not between P[0].X and P[N-1].X, ERR=2 and OUT contains the corresponding limiting value P[0].Y or P[N-1].Y.

If N lies outside of the allowed values which are between 2 and 11, then ERR=4.

Example in ST:

First of all ARRAY P must be defined in the header:

```
VAR
    ...
    CHARACTERISTIC_LINE:CHARCURVE;
    KL:ARRAY[0..10] OF POINT:=(X:=0,Y:=0),(X:=250,Y:=50),
    (X:=500,Y:=150),(X:=750,Y:=400),7((X:=1000,Y:=1000));
    COUNTER:INT;
    ...
END_VAR
```

Then we supply CHARCURVE with for example a constantly increasing value:

```
COUNTER:=COUNTER+10;
CHARACTERISTIC_LINE(IN:=COUNTER,N:=5,P:=KL);
```

The subsequent tracing illustrates the effect:



RAMP_INT

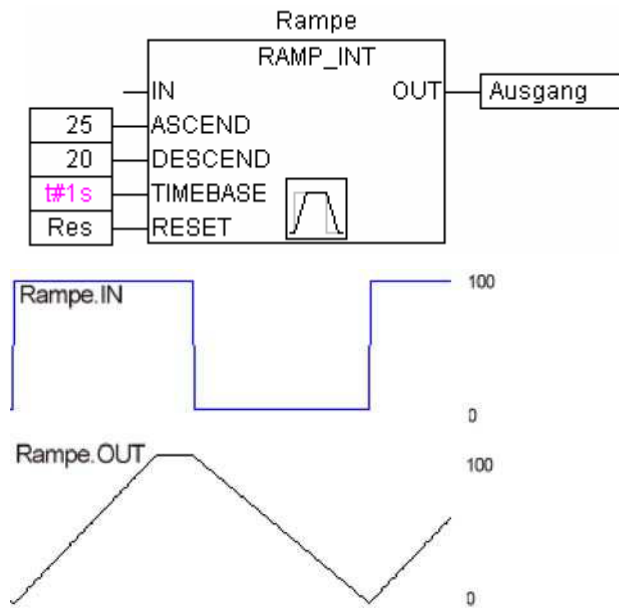
RAMP_INT serves to limit the ascendance or descendance of the function being fed:

The input consists on the one hand out of three INT values: IN, the function input, and ASCEND and DESCEND, the maximum increase or decrease for a given time interval, which is defined by TIMEBASE of the type TIME. Setting RESET to TRUE causes RAMP_INT to be initialised anew.

The output OUT of the type INT contains the ascend and descend limited function value.

When TIMEBASE is set to t#0s, ASCEND and DESCEND are not related to the time interval, but remain the same.

Example in CFC:



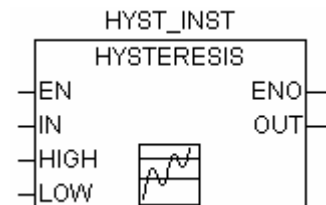
RAMP_REAL

RAMP_REAL functions in the same way as RAMP_INT, with the simple difference that the inputs IN, ASCEND, DESCEND and the output OUT are of the type REAL.

10.18.7 Analog Value Processing...

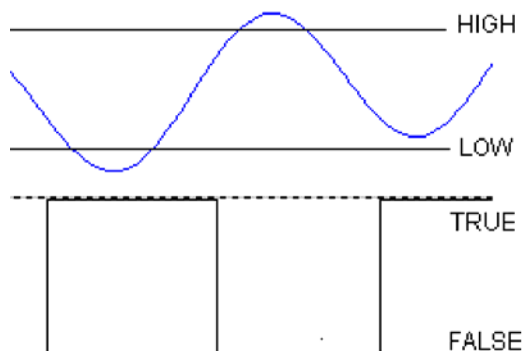
HYSTERESIS

The input to this function block consists of three INT values IN, HIGH and LOW. The output OUT is of the type BOOL.



If IN goes below the limiting value LOW, OUT becomes TRUE. If IN goes over the upper limit HIGH, FALSE is delivered.

An illustrative example:



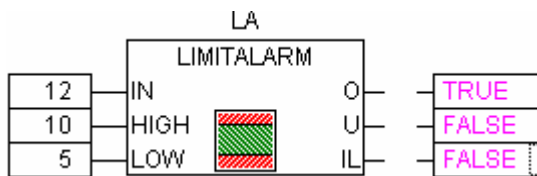
LIMITALARM

This function block specifies whether the input value is within a set range and which limits it has violated if it has done so.

The input values IN, HIGH and LOW are each of the type INT, while the outputs O, U and IL are of the type BOOL.

If the upper limit HIGH is exceeded by IN, O becomes TRUE, and when IN is below LOW, U becomes TRUE. IL is TRUE if IN lies between LOW and HIGH.

Example in FBD: Result:



10.19 The AnalyzationNew.lib library

This library provides modules for the analysis of expressions. If a composed expression is FALSE, those of its components can be evaluated which are adding to this result. In the SFC-Editor the flag SFCErrAnalyzationTable uses this function implicitly for the analysis of expressions in transitions.

Example of an expression:

```
b OR NOT(y < x) OR NOT (NOT d AND e)
```

The functions:

The following variables are used by all modules:

InputExpr: BOOL, expression to be analysed

DoAnalyze: BOOL, TRUE starts analysis

ExpResult: BOOL, current value of the expression

Different is the output of the result of the analyzation:

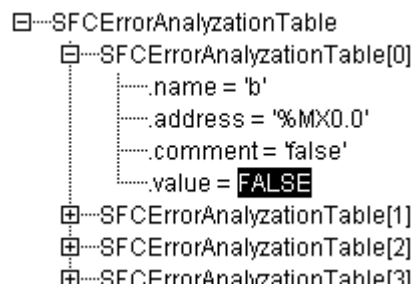
AnalyseExpression returns in a string the components of the expression, which are adding to the total value FALSE. Function **AppendErrorString** is used for this purpose, separating the particular components in the output string by "|" characters.

OutString: STRING, Result of the analysis, Sequence of the concerned components of the expression (e.g. y < x | d)

AnalyseExpressionTable writes the components of the expression, which are adding to the total value FALSE, to an array. For each component the following information is provided by **structure ExpressionResult**: name, address, comment, (current)value.

OutTable: ARRAY [0..15] OF ExpressionResult;

e.g.



AnalyseExpressionCombined combines the functionalities of AnalyseExpression plus AnalyseExpressionTable

10.20 CoDeSys System Libraries

Note

It depends on the currently used target system which system libraries are supported and can be used in the program. For respective information please see the document SysLibs_Overview.pdf.

Appendix E: Operators and Library Modules Overview

The table shown below shows an overview on the **operators**, which are available in CoDeSys resp. in the libraries **Standard.lib** and **Util.lib**. You find there the notation for ST and IL. For IL also the supported modifiers are listed.

Take note that for the 'IL operator' column: Only the line in which the operator is used will be displayed. A prerequisite is that the (first) required operand have been successfully loaded in the preceding line (e.g. LD in).

The '**Mod. IL**' column shows the possible modifiers in IL:

- C** The command is only executed if the result of the preceding expression is TRUE.
- N** for JMPC, CALC, RETC: The command is only executed if the result of the preceding expression is FALSE.
- N** otherwise: negation of the operand (not of the accumulator)
- (** Operator enclosed in brackets: only after the closing bracket is reached will the operation preceding the brackets be carried out.

Please obtain a detailed description of usage from the appropriate Appendices concerning IEC operators integrated into CoDeSys resp. the libraries.

10.21 Operators in CoDeSys

<i>in ST</i>	<i>in AWL</i>	<i>Mod. AWL</i>	<i>Description</i>
'			String delimiters (e.g. 'string1')
..			Size of Array range (e.g. ARRAY[0..3] OF INT)
[]			
:			Delimiter between Operand and Type in a declaration (e.g. var1 : INT;)
;			Termination of instruction (e.g. a:=var1;)
^			Dereferenced Pointer (e.g. pointer1^)
	LD var1	N	Load value of var1 in buffer
:=	ST var1	N	Store actual result to var1
	S boolvar		Set boolean operand boolvar exactly then to TRUE, when the actual result is TRUE
	R boolvar		Set boolean operand boolvar exactly then to FALSE, when the actual result is TRUE
	JMP label	CN	Jump to label
<Program name>	CAL prog1	CN	Call program prog1
<Instance name>	CAL inst1	CN	Call function block instance inst1
<Fctname>(vx, vy,..)	<Fctname> vx, vy	CN	Call function fctname and transmit variables vx, vy
	(The value following the bracket is handled as operand, the operation before the bracket is not executed before the expression in the brackets.

<i>in ST</i>	<i>in AWL</i>	<i>Mod. AWL</i>	<i>Description</i>
)		Now execute the operation which has been set back
AND	AND	N,(Bitwise AND
OR	OR	N,(Bitwise OR
XOR	XOR	N,(Bitwise exclusive OR
NOT	NOT		Bitwise NOT
+	ADD	(Addition
-	SUB	(Subtraction
*	MUL	(Multiplication
/	DIV	(Division
>	GT	(Greater than
>=	GE	(Greater or equal
=	EQ	(Equal
<>	NE	(Not equal
<=	LE	(Less or equal
<	LT	(Less than
MOD(in)	MOD		Modulo Division
INDEXOF(in)	INDEXOF		Internal index of POU in1; [INT]
SIZEOF(in)	SIZEOF		Number of bytes required for the given data type of in
SHL(K,in)	SHL		Bitwise left-shift of operator in by K
SHR(K,in)	SHR		Bitwise right-shift of operator in by K
ROL(K,in)	ROL		Bitwise rotation to the left of operator in by K
ROR(K,in)	ROR		Bitwise rotation to the right of operator in by K
SEL(G,in0,in1)	SEL		Binary selection between 2 operands in0 (G is FALSE) and in1 (G is TRUE)
MAX(in0,in1)	MAX		Returns the greater of 2 values
MIN(in0,in1)	MIN		Returns the lesser of 2 values in0 and in1
LIMIT(MIN,in,Max)	LIMIT		Limits the value range (in is set back to MIN or MAX in case of exceeding the range)
MUX(K,in0,...in_n)	MUX		Selects the Kth value out of a group of values (in0 to In_n)
ADR(in)	ADR		Address of the operand in [DWORD]
ADRINST()	ADRINST()		Address of the function block instance from which you are calling that operator.
BITADR(in)	BITADR		Bit-Offset des Operanden in [DWORD]
BOOL_TO_<type>(in)	BOOL_TO_<type>		Type conversion of the boolean operand

<i>in ST</i>	<i>in AWL</i>	<i>Mod. AWL</i>	<i>Description</i>
<type>_TO_BOOL (in)	<type>_TO_BOOL		Type conversion to BOOL
INT_TO_<type> (in)	INT_TO_<type>		Type conversion of an INT Operand to another elementary type
REAL_TO_<type> (in)	REAL_TO_<type>		Type conversion of an REAL operand to another elementary type
LREAL_TO_<type> (in)	LREAL_TO_<type>		Type conversion of a LREAL operand to another elementary type
TIME_TO_<type> (in)	TIME_TO_<type>		Type conversion of a TIME operand to another elementary type
TOD_TO_<type> (in)	TOD_TO_<type>		Type conversion of a TOD operand to another elementary type
DATE_TO_<type> (in)	DATE_TO_<type>		Type conversion of a DATE operand to another elementary type
DT_TO_<type> (in)	DT_TO_<type>		Type conversion of a DT operand to another elementary type
STRING_TO_<type> (in)	STRING_TO_<type>		Type conversion of a string operand to another elementary type, in must contain valid value of desired type
TRUNC (in)	TRUNC		Conversion from REAL to INT
ABS (in)	ABS		Absolute value of operand in
SQRT (in)	SQRT		Square root of operand in
LN (in)	LN		Natural logarithm of operand in
LOG (in)	LOG		Logarithm of operand in, base 10
EXP (in)	EXP		Exponential function of operand in
SIN (in)	SIN		Sine of operand in
COS (in)	COS		Cosine of operand in
TAN (in)	TAN		Tangent of operand in
ASIN (in)	ASIN		Arc sine of operand in
ACOS (in)	ACOS		Arc cosine of operand in
ATAN (in)	ATAN		Arc tangent of operand in
EXPT (in,expt)	EXPT expt		Exponentiation of operand in with expt

10.22 Elements of the Standard.lib:

<i>in ST</i>	<i>in AWL</i>	<i>Description</i>
LEN (in)	LEN	String length of operand in
LEFT (str,size)	LEFT	Left initial string of given size of string str
RIGHT (str,size)	RIGHT	Right initial string of given size of string str
MID (str,size,pos)	MID	Partial string of str of given size at position pos
CONCAT ('str1','str2')	CONCAT 'str2'	Concatenation of two subsequent strings
INSERT ('str1','str2',pos)	INSERT 'str2',p	Insert string str1 in String str2 at position pos
DELETE ('str1',len,pos)	DELETE len,pos	Delete partial string (length len), start at position pos of str1
REPLACE ('str1','str2',len,pos)	REPLACE 'str2',len,pos	Replace partial string of length len by str2, start at position pos of str1
FIND ('str1','str2')	FIND 'str2'	Search for partial string str2 in str1
SR	SR	Bistable FB is set dominant
RS	RS	Bistable FB is set back
SEMA	SEMA	FB: Software Semaphor (interruptable)
R_TRIG	R_TRIG	FB: rising edge is detected
F_TRIG	F_TRIG	FB: falling edge is detected
CTU	CTU	FB: Counts upv
CTD	CTD	FB: Counts down
CTUD	CTUD	FB: Counts up and down
TP	TP	FB: trigger
TON	TON	FB: Timer On-Delay
TOF	TOF	FB: Timer Off-Delay
RTC	RTC	FB: Real Time Clock

10.23 Elements of the Util.lib

BCD_TO_INT	Conversion of a Byte: BCD to INT format
INT_TO_BCD	Conversion of a Byte: INT to BCD format
EXTRACT (in,n)	The n-th bit of DWORD in is returned in BOOI
PACK	Up to 8 bits are packed into a byte
PUTBIT	A bit of a DWORD is set to a certain value
UNPACK	A Byte is returned as single bits
DERIVATIVE	Local derivation

INTEGRAL	Integral
LIN_TRAFO	Transformation of REAL values
STATISTICS_INT	Min.,Max, Average values in INT format
STATISTICS_REAL	Min.,Max, Average in REAL format
VARIANCE	Variance
PD	PD controller
PID	PID controller
BLINK	Pulsating signal
FREQ_MEASURE	Measuring frequency of boolean input signal
GEN	Periodic functions
CHARCURVE	Linear functions
RAMP_INT	Limiting ascendance of descendance of the function beeing fed (INT)
RAMP_REAL	Limiting ascendance of descendance of the function beeing fed (REAL)
HYSTERESIS	Hysteresis
LIMITALARM	Watches whether input value exceeds limits of a defined range

Appendix F: Command Line-/Command File

10.24 Command Line Commands

When CoDeSys is started, you can add commands in the command line which will be asserted during execution of the program. These commands start with a "/". Capitalization/Use of small letters is not regarded. The commands will be executed sequentially from the left to the right.

/online	Immediately after start CoDeSys tries to go online with the current project.
/batch	CoDeSys starts without user interface and returns the error code of the first error resp. the return value of the first command which is terminated with a warning. CoDeSys will terminate immediately after the command file has been processed. The processing of the command file will be aborted as soon as the first command is processed with an error. Warnings do not terminate the processing. If neither errors nor warnings occur, the return value is S_OK. The return value always is coded as HRESULT.
/run	After login CoDeSys starts the application program. Only valid in combination with /online.
/show ... /show hide /show icon /show max /show normal	Settings for the CoDeSys frame window can be made. The window will not be displayed, it also will not be represented in the task menu. The window will be minimized in display. The window will be maximized in display. The window will be displayed in the same status as it was during the last closing.
/out <outfile>	All messages are displayed in the message window and additionally are written in the file <outfile>.
/noinfo	No splash screen at start of CoDeSys
/userlevel <group>	Definition of the user group (e.g. "/userlevel 0" for user group 0)
/password <password>	Direct input of the user group password (e.g. "/password abc")
/openfromplc	The project which is currently available on the connected target system, will be loaded.
/visudownload	If CoDeSys HMI is started with a project, which does not match with the project currently available on the target system, a download will be offered. (Dialog, to be closed with YES or NO).
/notargetchange	A change of the target system only can be done via a command file. See chapter 10.25, command "target...".
/cmd <cmdfile>	After starting the commands of the <cmdfile> get executed.

Regard the following syntax for a command line:

```
"<Path of CoDeSys-exe-file>" "<Path of the project>" /<command1> /<command2> ....
```

Example for a command line:

The project ampel.pro gets opened, but no window opens. The commands included in the command file command.cmd will be executed. Put double quotation marks around a path.

```
"D:\dir1\codesys" "C:\projects\ampel.pro" /show hide /cmd command.cmd
```

10.25 Command File (cmdfile) Commands

See the following table for a list of commands, which can be used in a command file (<cmdfile>). The command file you then can call by a command line (see above). There is no case sensitivity. The command line will be displayed as a message in the message window and can be given out in a message file (see below) except the command is prefixed by a "@".

All signs after a semicolon (;) will be ignored (comment). Parameters containing blanks must be embraced by quotation marks. Umlauts only may be used if the command file is created in ANSI code. Keywords can be used in the command parameters. A list of the keywords you find subsequent to the following tables of command descriptions.

Commands for controlling the subsequent commands:

onerror continue	The subsequent commands will be executed even if an error occurs.
onerror break	The subsequent commands will not be executed any more if an error has been detected.

Commands of the online menu:

online login	Login with the loaded project ('Online' 'Login')
online logout	Logout ('Online' 'Logout')
online run	Start of the application program ('Online' 'Run')
online stop	Stop application program ('Online' 'Stop')
online bootproject	Creation of a boot project. This command can be applied in offline and online mode ! (See also chapter 4.6description on command ,Online' ,Create Bootproject'!)
online sourcecodedownload	Download of the source code of the project to the PLC ('Online' 'Sourcecode download')
online sim	Switch on of simulation mode ('Online' 'Simulation')
online sim off	Switch off of simulation mode ('Online' 'Simulation')

Commands of the file menu:

file new	A new project is created ('File' 'New')
file open <projectfile> possible additions: /readpwd:<readpassword> /writepwd:<writepassword>	The project <projectfile> will be loaded ('File' 'Open') The password for read access is given here so that no dialog asking for the password will appear when the read-protected project is opened. The password for full access is given here, so that no dialog asking for the password will appear when the project is opened.
file close	The current project will be closed ('File' 'Close')
file save	The current project will be stored ('File' 'Save')
file saveas <projectfile> optionally add: <type><version>	The current project will be saved with the file name <projectfile> ('File' 'Save as') Default: Project will be saved as <projectfile>.pro under the current CoDeSys version. If you want to save the project as an internal or external library or as project for an older CoDeSys version, add the

	<p>respective command:</p> <p>Possible entries for <type>:</p> <p>"internallib" Save as internal library:</p> <p>"externallib" Save as external library:</p> <p>"pro" Save as project for older version:</p> <p>valid entries for <Version>: 15, 20, 21, 22 (product versions 1.5, 2.0, 2.1, 2.2)</p> <p>Example: "file save as lib_xy internallib22" -> The project "project xy.pro", which is created in the current CoDeSys Version will be saved as "lib_xy.lib" for V2.2.</p>
file saveas <projectfile>	The current project will be saved with the file name <projectfile> ('File' 'Save as')
file printersetup <filename>.dfr optionally add: pageperobject or pagepersubject	Define a document frame file ('File' Printer setup') and optionally define one of the print options 'New page per object' or 'New page per subobject' ; these settings affect the printing of the document (project documentation, see below)
file archive <filename>.zip	The project will be archived in a zip-file with the given filename ('File' Save/Mail Archive')
file quit	CoDeSys will be closed ('File' 'Exit')

Commands of the project menu:

project build	The project that is loaded will be incrementally compiled ('Project' 'Build')
project rebuild or project compile	The project that is loaded will be compiled in full ('Project' 'Rebuild')
project clean	Compilation information and Online Change information in the current project will be deleted ('Project' 'Clean Project')
project check	The project that is loaded will be checked ('Project' 'Check all')
project compile	The current project will be compiled by "Rebuild all" ('Project' 'Rebuild all')
project check	The current project will be checked ('Project' 'Check')
project build	The current project will be built ('Projekt' 'Build')
project import <file1> ... <fileN>	The files <file1> ... <fileN> get imported into the current project ('Project' 'Import'). Regard: Wildcards can be used, e.g. "project import C:\projects*.exp" will import all files with extension *.exp found in directory C:\projects.
project export <expfile>	The current project will be exported in the file <expfile> ('Project' 'Export')
project expmul	Each object of the current project will be exported in an own file, which gets the name of the object.
project documentation	The entire project will be printed on the default printer ('Project' 'Documentation', see also above "file printersetup")

Commands for the control of the message file:

out open <msgfile>	The file <msgfile> opens as message file. New messages will be appended
out close	The currently shown message file will be closed.
out clear	All messages of the currently opened message file will be deleted.

Commands for the control of messages:

echo on	The command lines will be displayed as messages.
echo off	The command lines will not be displayed as messages.
echo <text>	<text> will be displayed in the message window.

Commands for the control of replace of objects respectively for the control of files for import, export, copy:

replace yesall	Replace all (any 'query on' command will be ignored; no dialogs will open)
replace noall	Replace none (any 'query on' command will be ignored; no dialogs will open)
replace query	If a 'query on' command is set, then a dialog will open regarding the replacing of the objects even if there is a 'replace yesall' or 'replace noall' command

Commands for the control of the default parameters of CoDeSys dialogs:

query on	Dialogs are displayed and need user input
query off ok	All dialogs respond as if the user had clicked on the 'OK' button
query off no	All dialogs respond as if the user had clicked on the 'No' button
query off cancel	All dialogs respond as if the user had clicked on the 'Cancel' button

Command for calling command files as subprograms:

call <parameter1> ... <parameter10>	Command files will be called as subprograms. Up to 10 parameters may be passed. In the file that is called, the parameters can be accessed with \$0 - \$9.
call <parameter1> ... <parameter10>	Command files are called as subroutines. Up to ten parameters can be consigned. In the subroutine called you can access the parameters using \$0 - \$9.

Setting of directories used by CoDeSys (-> project options dialog, category 'Directories', subcategory 'General'): If several directories are defined with one of the following commands, these must be separated by a semicolon + empty space and the whole row of directories must be embraced by double quotation marks. Example, two paths:

```
dir lib "D:\codesys\Libraries\Standard; D:\codesys\Libraries\NetVar"
```

dir lib <libdir>	Sets <libdir> as the library directory
dir compile <compiledir>	Sets <compiledir> as the directory for the compilation files
dir config <configdir>	Sets < configdir > as the directory for the configuration files
dir upload <uploaddir>	Sets < uploaddir > as the directory for the upload files

Delaying processing of the CMDFILE:

delay 5000	Waits 5 seconds
-------------------	-----------------

Controlling the Watch and Receipt Manager:

watchlist load <file>	Loads the Watchlist saved as <file> and opens the corresponding window ('Extras' 'Load Watchlist')
watchlist save <file>	Saves the current Watchlist as <file> ('Extras' 'Save Watchlist')
watchlist set <text>	The watchlist is set active (corresponds to the selection of a list in the left part of the watch and receipt manager window)
watchlist read	Updates the values of the Watch variables ('Extras' 'Read receipt')
watchlist write	Fills the Watch variables with the values found in the Watchlist ('Extras' 'Write receipt')

Linking libraries:

library add <library file1> <library file2> .. <library fileN>	Attaches the specified library file to the library list of the currently open project. If the file path is a relative path, the library directory entered in the project is used as the root of the path.
library delete [<library1> <library2> .. <libraryN>]	Deletes the specified libraries from the library list of the currently open project.

Copying objects:

object copy <source project file> <source path> <target path>	<p>Copies objects from the specified path of the source project file to the target path of the already opened project.</p> <p>If the source path is the name of an object, this will be copied.</p> <p>If it is a folder, all objects below this folder will be copied. In this case, the folder structure below the source folder will be duplicated.</p> <p>If the target path does not yet exist, it will be created.</p>
--	--

Read-only access for particular objects :

object setreadonly <TRUE FALSE> <object type> <object name>	<p>Sets read-only access to a object; Define the object type and in case of object types pou, dut, gvl, vis also the name of the object.</p> <p>Possible object types: pou, dut (data type), gvl (global variables list), vis (visualization), cnc (CNC object), liblist (Libraries), targetsettings, toolinstanceobject (particular Tools instance), toolmanagerobject (all instances in the Tools tree), customplconfig (PLC configuration), projectinfo (Project information), taskconfig (task configuration), trace, watchentrylist (Watch- and Recipe Manager), alarmconfig (Alarm configuration)</p> <p>e.g. "object setreadonly TRUE pou plc_prg" will set the PLC_PRG to read-only access</p>
--	--

Entering communication parameters (gateway, device):

gateway local	Sets the gateway on the local computer as the current gateway.
gateway tcpip <Address> <Port>	<p>Sets the gateway in the specified remote computer as the current gateway.</p> <p><Address>: TCP/IP address or hostname of the remote computer</p> <p><Port>: TCP/IP port of the remote gateway</p> <p>Important: Only gateways that have no password set can be reached!</p>

device guid <guid>	Sets the device with the specified GUID as the current device. GUID must have the following format: {01234567-0123-0123-0123-0123456789ABC} The curly brackets and the hyphens must appear at the specified positions.
device instance <Instance name>	Sets the instance name for the current device to the name specified
device parameter <Id> <Value>	Assigns the specified value, which will then be interpreted by the device, to the parameter with the specified ID.

System call:

system <command>	Carries out the specified operating system command.
-------------------------------	---

Select target system:

target <Id>	Sets the target platform for the current project. If CoDeSys is getting started with command line option "/notargetchange" (see Chapter 10.24), only by this command a target can be set.
--------------------------	---

Query system state:

state offline	Returns "S_OK", if currently there is no connection between programming system and target system (offline mode), otherwise "HRESULT[0x800441f0]" (online mode)..
state online	Returns "S_OK", if currently there is a connection between programming system and target system (online mode), otherwise "HRESULT[0x800441f0]" (offline mode).

Password for user group:

With the following settings - when opening a project which is protected by user group passwords - you can enter the password for a certain user group. Thus, even if the display of the user input dialogs is switched off ("query off...", see above), a password-protected project can be opened via the command file. The entries for the user group and the password must be placed before the command "file open...!"

Example:

```
user level 0
user password aaa
file open "D:\codesys\projects\xxxx.pro"
query off ok
```

user level	User group, the password for which is defined in the project and is given by the subsequent command "user password".
user password	Password for user group specified by the preceding command "user group".

Visualization settings:

visual settings...	corresponds to the possible settings which can be done for a visualization in 'Extras' 'Settings', category Language, resp. in the Target Settings, category Visualization.
... language file on off	Option 'Language file' gets activated (on) or deactivated (off). In case of activation the option 'Dynamic texts' will be deactivated.

... set languagefile <Dateipfad Sprachdatei>	Specification of the language file to be used (.tlt or .vis). Example: "visual settings set languagefile proj1.tlt.
... dynamictexts on off	Option 'Dynamic texts' will be deactivated (on) or deactivated (off). In case of activation the option 'Language file' will be deactivated.
... dynamictextfiles <Dateipfad> <Dateipfad> ...	Specification of a list of language file paths to be used. The previous list will be deleted. Example: "visual settings D:\dynfiles\p1.xml D:\dynfiles\p2.xml"
... dynamictexthideelements on off	Activation resp. deactivation of option 'Suppress elements if no text replacement has taken place'.
... language <Sprache>	Specification of the language to be used; Example: "visual settings language German""
... tablekeyboardusage_web on off	Activation resp. deactivation of option 'Keyboard usage for tables' in the Web-Visualization (Target Settings). "
... tablekeyboardusage_codesys on off	Activation resp. deactivation of option 'Keyboard usage for tables' in CoDeSys resp. CoDeSys-HMI (Target Settings).
visual webvisuactivation on off	Activation resp. deactivation of option 'Web-Visualization' (Target Settings)).).

Commands concerning managing the project in the ENI project data base:

In the following in the description of the commands placeholders are used:

<category>: Replace by "project" or "shared" or "compile" depending on which of the following data base categories is concerned: Project Objects, Shared Objects, Compile Files

<POUname>: Name of the object, corresponds to the object name which is used in CoDeSys.

<Objecttype>: Replace by the shortcut, which is appended as an extension to the POU name of the object in the data base, and which reflects the object type (defined by the list of object types, see ENI Administration, 'Object Types').

Example: Object "GLOBAL_1.GVL" -> the POU name is "GLOBAL_1", the object type is "GVL" (global variables list)

<comment>: Replace by a comment text (embraced by single quotation marks), which will be stored in the version history with the particular action.

Commands to configurate the project data base link via the ENI Server:

eni on eni off	The option 'Use source control (ENI)' will be activated resp. deactivated (Dialog 'Project' 'Options' 'Project source control')
eni project readonly on eni project readonly off	The option 'Read only' for the data base category 'Project objects' will be activated resp. deactivated (Dialog 'Project' 'Options' 'Project objects')
eni shared readonly on eni shared readonly off	The option 'Read only' for the data base category 'Shard objects' will be activated resp. deactivated (Dialog 'Project' 'Options' 'Shared objects')
eni set local <POUname>	The object will be assigned to category 'Local', i.e. it will not be stored in the project data base (Dialog 'Project' 'Object' 'Properties' 'Data base-connection')
eni set shared <POUname>	The object will be assigned to category 'Shared objects' (Dialog 'Project' 'Object' 'Properties' 'Data base-connection')
eni set project <POUname>	The object will be assigned to category 'Project objects'

	(Dialog 'Project' 'Object' 'Properties' 'Data base-connection')
eni <category> server <TCP/IP_Address> <Port> <Projectname> <Username> <Password>	Configures the connection to the ENI Server for the category 'Project objects' (Dialog 'Project' 'Options' 'Project data base'); Example: eni project server localhost 80 batchtest\project EniBatch Batch (TCP/IP-Address = localhost, Port = 80, Project name = batchtest\project, User name = EniBatch, Password = Batch)
eni compile sym on eni compile sym off	The option 'Create ASCII symbol information (.sym)' for the objects of category 'Compile files' will be activated/deactivated (Dialog 'Project' 'Options' 'Project source control' 'ENI settings' for 'Compile files')
eni compile sdb on eni compile sdb off	The option 'Create binary symbol information (.sym)' for the objects of category 'Compile files' will be activated/deactivated (Dialog 'Project' 'Options' 'Project source control' 'ENI settings' for 'Compile files')
eni compile prg on eni compile prg off	The option 'Create boot project' for the objects of category 'Compile files' will be activated/deactivated (Dialog 'Project' 'Options' 'Project source control' 'ENI settings' for 'Compile files')

Commands of the menu 'Project' 'Data Base Link' for working with the data base:

eni set <category>	The object gets assigned to the named data base category ('Define')
'eni set <category>set <Objecttype>:<POUname> <Objecttype>:<POUname>	The objects which are listed separated by spaces will be assigned to the named data base category. ('Multiple Define') Example: "eni set project pou:as_fub pou:st_prg" -> the objects (pou) as_fub and st_prg get assigned to category 'Project objects'
eni <category> getall	The latest version of all objects of the named category will be called from the data base ('Get All Latest Versions')
'eni <category>get <Objecttype>:<POUname> <Objecttype>:<POUname>	The objects of the named category, which are listed separated by spaces will be called from the data base. ('Multiple Define'). ('Get latest version') Example: "eni project get pou:as_fub gvl:global_1" -> the POU as_fub.pou and the global variables list global_1.gvl will be called from the data base
eni <category> checkoutall "<comment>"	All objects of the named category will be checked out from the data base. The defined comment will be stored with the check-out-action in the version history.
eni <category> checkout "<comment>" <Objecttype>:<POUname> <Objecttype>:<POUname>	All objects (Objecttype:POUname) which are listed separated by spaces will be checked out from the data base. The defined comment will be stored with the check-out-action in the version history for each particular object. Example: "eni project checkout "for working on xy" pou:as_fub gvl:global_1" ➔ The POU as_fub and the global variables list global_1 will be checked out and the comment "for working on xy" will be stored with this action

	➔
eni <category>checkinall "<comment>"	All objects of the project, which are under source control in the project data base, will be checked in. The defined comment will be stored with the check-in-action.
eni <category> checkin "<comment>" <Objecttype>:<POUname> <Objecttype>:<POUname>	All objects (Objecttype:POUname) which are listed separated by spaces will be checked in to the data base. The defined comment will be stored with the check-in-action in the version history for each particular object. (see above: check out) The defined comment will be stored with the check-in-action in the version history for each particular object.

Keywords for the command parameters:

The following keywords, enclosed in "\$", can be used in command parameters:

\$PROJECT_NAME\$	Name of the current CoDeSys project (file name without extension ".pro", e.g. "project_2.pro")
\$PROJECT_PATH\$	Path of the directory, where the current CoDeSys project file is (without indication of the drive and without a backslash at the end, e.g. "projects\sub1").
\$PROJECT_DRIVE\$	Drive, where the current CoDeSys project is (without backslash at the end, e.g. "D:")
\$COMPILE_DIR\$	Compile directory of the current CoDeSys project (with indication of the drive and without backslash at the end, e.g. "D:\codesys\compile")
\$EXE_DIR\$	Directory where the codesys.exe file is (with indication of the drive and without backslash at the end, e.g. D:\codesys)

Example of a command file:

A command file like shown below will open the project file ampel.pro, will then load a watch list, which was stored as w.wtc, will then start the application program and write – after 1 second delay - the values of the variables into the watch list watch.wtc (which will be saved) and will finally close the project.

```
file open C:\projects\CoDeSys_test\ampel.pro
query off ok
watchlist load c:\work\w.wtc
online login
online run
delay 1000
watchlist read
watchlist save $PROJECT_DRIVE$\$PROJECT_PATH$w_update.wtc
online logout
file close
```

This command file will open the project ampel.pro, will load an existing watchlist w.wtc, will start the application program, after 1 second will write the variables values to the watch list w_update.wtc, which will be saved in the directory "C:\projects\CoDeSys_test" and then will close the project again.

A command file is called in a command line like shown here:

```
"<path of codesys.exe>" /cmd "<path of cmd file>"
```


Appendix G: Siemens Import

In the **'Project' 'Siemens Import'** submenu, you will find commands which allow you to import POU's and variables from Siemens STEP5 files. The command "Import from a SEQ symbol file" allows you to import global variables from STEP5 symbol files. Run this command before either the command **'Import from a S5 project file'** so that readable symbol names can be created when the POU's are imported. These two commands allow you to import POU's from STEP5 program files. When this done, the POU's are inserted into the open CoDeSys project. You can select whether the POU's will remain in the STEP5 IL language or be converted to an IEC language.

We recommend that the CoDeSys project into which you are importing be empty. Of course, you must be certain that the library standard.lib is linked to your project, otherwise you will be unable to import the counter and the timer.

10.26 Import from a SEQ Symbol File

SEQ format is a common format for symbol files in a STEP5 project. Symbol assignments can be read from SEQ symbol files (*.seq). A symbol assignment contains an absolute address for a S5 program element (input, output, memory location, etc.), a corresponding symbol identifier and may also contain comments about the symbol. A SEQ file is text file that contains one assignment of this type per line. Each of the "Fields" in the line are separated by Tabs. Also each line can only hold one comment which must begin with a semicolon.

The symbol assignments in the SEQ file will be translated into global variable declarations based on IEC 61131-3. The symbolic name, the address and the comment (if available) will be transferred during this process. The address will be adapted to IEC 61131-3 (Percent sign, etc.). Since a S5 symbol name can contain characters that are not permitted in an IEC identifier, the names will be changed if necessary. Invalid characters will be replaced by the underscore character. Should there be more than one underscore character in a row, every second one would be replaced by a valid character (e.g., "0"). If a symbol name is changed during the conversion, the original name will be added in a comment after the change. SEQ comment lines will be transferred as comments. Multiple blocks of global variables can be created. Each block consists of less than 64K of text.

The SEQ format described is used in Siemens STEP5-PG Software, in most versions of the Siemens STEP7-300/400 and in ACCON-PG from DELTALOGIC. This format is supported in STEP7-SEQ files created in version 3.x or better. STEP7 version 2.x can export a different SEQ format that is not supported. Instead of using separators (Tabs), it is based on a fixed length for the symbolic name and uses blanks if necessary.

You first select the SEQ file in a standard Windows dialog box. Then perform the import, when this is done the global variable list will be compiled. Errors may arise during this process when STEP5/7 identifiers are converted into IEC61131-3 compatible identifiers. For example, both STEP5 identifiers "A!" and "A?" would be converted into the IEC identifier "A_". At this point the following message would appear, "Multiple declarations with the same identifier A_". Change one of the variables.

Under absolutely no other circumstances should you make any changes to the global variable list. If you identify addresses that are valid in a Siemens PLC but are invalid in your Controller: Leave them alone for now, even if you get a thousand error messages while compiling. The addresses are needed exactly as they are in order to import the POU's.

If the project into which you are importing already contains a declaration for a global variable x with its address (e.g., "%MX4.0"), you may find that the SEQ import contains a variable defined with the same address. This is allowed in IEC 61131-3 but is generally not in the best interest of the user. No error message will appear, but your program may not function as it should since the address will be used in different POU's with no contextual reference. To avoid this problem, it is best to import into an empty project or into a project in which no absolute addresses have been used up to this point.

STEP5/7 Program Organization Units can be imported, once the SEQ import has been performed. You can also add the inputs and outputs that will be used in the PLC Configuration. These are not required for the STEP5/7 import but the addresses will be checked and may show up as errors when you rebuild the project.

10.27 Import from a S5 Project File

POUs can read from Siemens S5 program files (*.s5d). The code that it uses is MC5 Code that can be run by S5 SPS. In general, MC5 Code corresponds with the STEP5 Instruction List (without symbol names) with which the programmer is familiar. The S5D also contains the line comments from the STEP5 Instruction List. Since an S5D file contains only absolute addresses with no symbol names, CoDeSys searches for the symbol names among the current CoDeSys project variables. If none are found, the absolute address is left unchanged. Therefore, if you feel the symbol name is useful, import the SEQ file before the S5 file.

You first select the S5D file in a standard Windows dialog box. Another box pops up which contains the list of POU's from which you can select. It is best to select all of them. You can also select to leave the POU's in the STEP5 IL language or to convert them to IL, LD or FBD.

Symbol names will be used in place of absolute names as much as possible. If CoDeSys finds the instruction "U M12.0" during the import, it will search for a global variable set at memory location M12.0. The first declaration that fits this description will be taken and the instruction will be imported as "U-Name" instead of "U M12.0" (the name of the identifier for the memory location is M12.0).

At times additional variables may be needed during an import or code conversion. These additional variables will be declared globally. For example, R_TRIG instances are needed to reproduce edge-triggered inputs (e.g., in a S5 counter).

10.28 Converting S5 to IEC 61131-3

If you select an IEC language as your target language for a STEP5 import, you must be aware that portions of your project cannot be converted into IEC 61131-3. If part of a S5 POU contains code that cannot be converted into IEC 61131-3, an error message will be generated and the critical portion of the original STEP5 IL code will be inserted as a comment in the IEC POU. You must then replace or rewrite this code. System commands that only function in a specific S5 CPU cannot be converted into IEC. The "STEP5 Core Command List" can be converted into IEC code with a click of a button despite the fact that STEP5 is enormously different in the way it is conceived.

The core command list that can be converted to IEC 61131-3 contains all commands that can be converted to LD or FBD in a STEP5 programming system and also all commands that are allowed in a STEP5-PB (Program Block). In addition, of the STEP5 commands allowed only in IL or in FB's (function blocks), the commands that can be converted to IEC are primarily those that are available in every S5 CPU (e.g., absolute and conditional jumps, shift commands, etc.)

The only exception or limitation for conversion is related to resetting timers which can be done in STEP5 but not normally in IEC 61131-3.

The individual convertible commands:

U, UN, O, ON, S, R, = with the following bit operands: I (inputs), O (outputs), M (memory locations), S (S memory locations), D (Data in data blocks)

U, UN, O, ON with the following operands: T (Timer), C (Counter)

S, R with the following operands: C

SU, RU, P, PN with the following operands: E, A, M, D

O, O(, U(,)

L, T with the following operand ranges: E, A, M, D, T, C, P (Periphery) and operand sizes: B (byte), W (word), D (double word), L (left byte), R (right byte)

L with the following constant formats: DH, KB, KF, KH, KM, KT, KZ, KY, KG, KC

SI, SE, SA with the following operands: T

ZV, ZR with the following operands: C

+, -, X, : with the following operands: F (fixed point number), G (floating point number)

+, - with the following operands: D (32 bit fixed point number)

!=, >, >, <, >=, <= with the following operands: F, D, G

ADD with the following operands: BF, KF, DH

SPA, SPB with the following operands: PB, FB (with most parameter types), SB

A, AX with the following operands: DB, DX

BE, BEA, BEB

BLD, NOP, ***

UW, OW, XOW

KEW, KZW, KZD

SLW, SRW, SLD, RRD, RLD

SPA=, SPB=

SPZ=, SPN=, SPP=, SPM=

TAK

D, I

Most of the formal operand commands

Unconvertible Commands

U, UN, O, ON, S, R, = with the following bit operands: Timer and counter bits (T0.0, C0.0)

L, T with the following operand ranges: Q (expanded periphery)

LC with the following operands: T, C

SV, SS, R, FR with the following operands: T

FR with the following operands: C

Formal operand commands for starting, resetting and releasing timers

All commands with operands from the ranges BA, BB, BS, BT (operating system data).

SPA, SPB with the following operands: OB (works only with certain S5's and certain OBs)

BA, BAB with the following operands: FX

E, EX with the following operands: DB, DX

STP, STS, STW

DEF, DED, DUF, DUD

SVW, SVD

SPO=, SPS=, SPR

AS, AF, AFS, AFF, BAS, BAF

ENT

SES, SEF

B with the following operands: DW, MW, BS

LIR, TIR, LDI, TDI, TNW, TXB, TXW

MAS, MAB, MSA, MSB, MBA, MBS

MBR, ABR

LRW, LRD, TRW, TRD

TSG

LB, TB, LW, TW with the following operands: GB, GW, GD, CB, CW, CD

ACR, TSC

BI

SIM, LIM

If you examine the commands that cannot be converted you will see that they are generally special commands that are only available on certain CPUs. The standard commands that cannot be converted to IEC are: loading BCD coded timer or counter values (LC T, LC C), timer types SV and SS, and resetting timers.

Data Blocks

STEP5 data blocks are converted into POU's (Program Organization Units) that have a header but no code. This is convenient if the data blocks are used as normal variable ranges but inconvenient if attempts have been made to manually implement concepts like instance data blocks in the STEP5 program.

Other Problems when Importing from STEP5

The STEP5 import can be improved manually in the following ways.

1. Time values in word variables

In STEP5 a time value is allowed in every word address be it in the memory location area or in a data block. This is not allowed in IEC 61131-3, TIME variables or constants are not compatible with WORD addresses. This can result in the creation of erroneous command sequences when importing from STEP5. This will not happen if you open a data block and select the time format (KT) for the address in question. In other words, this error only occurs when the STEP5 program is worth the effort of improving it. When it does occur, you will see the message "Incompatible Types: Cannot convert WORD to TIME." or "Incompatible Types: Cannot convert TIME to WORD." You must then modify the declaration for the WORD variable (if available) and turn it into a TIME variable.

2. Failure to Access Data Blocks

There are no data blocks in IEC 61131-3 and it is impossible completely to recreate them in IEC. In STEP5 they are used as normal variable ranges (almost like a memory location ranges), and also in the form of arrays (B DW), pointers (B MW100 A DB 0) or unions (byte, word or double word access in DBs). STEP5 conversion can only convert DB access if it is somewhat structured. When attempting to access DBs you must know which DB is open (A DB). You must be aware of this when the A DB operation is closer to the beginning in the same POU or when the DB number is included with the POU as a formal parameter. If A DB is not found in front of the first DB access, the POU cannot be converted. The warning "No open data block (insert an A DB)" notifies you that this is the case. In the converted POU, you will see access to an undefined variable named "ErrorDW0" (for example) that will cause an error message to be generated when the newly converted POU is compiled. You can then replace the variables with access to the correct DB (e.g., replace "ErrorDW0" with "DB10.DW0"). The other option is to discard the converted POU and insert an A DB at the beginning of the POU in STEP5.

A STEP5 POU that accesses data words (data bytes, etc.) should always open the data block first. If necessary, the POU should be improved before being imported by inserting the appropriate A DB command preferably at the beginning of the POU. Otherwise the converted POU will have to be edited after the fact.

If there is more than one A BD operation that must be partially skipped, the conversion may have a errors, i.e., code may be generated that accesses the wrong DB.

3. Higher Concepts Related to Data Block Access

In STEP5 you have the option of creating something similar to instances by having the Code block open an indexed version of a data block. This could be done with the following sample code sequence:

```
L KF +5
```

T MW 44

B MW 44

A DB 0

The DB5 is opened at the end of this sequence (in general, the DB whose number is found in the memory location word %MW44 will be opened). This type of access is not recognized in the conversion which means that the following changes have to be made after the conversion:

First, all DBs must be imported that act as instance DBs , e.g., DB5 and DB6. They will be imported as normal IL, LD or FBD POU's whichever you prefer. The POU's do not have a code, but rather a header with definitions of local variables. Type instances can now be created from these POU's. Create a user-defined type (e.g., named DBType) and insert the local variables and converted DBs as components. Then create global instances of this type by writing to a global variable list:

```
VAR_GLOBAL
  DB5, DB6 : DBType;
END_VAR
```

You can now delete the converted DBs from your project.

Then you have to create a copy of the indexed version of the DBs by giving the corresponding POU another VAR_INPUT parameter of the type DBType. Data access within the POU must now be redirected to this instance. You must then include one of the instance DBs as an actual parameter when you open it.

4. The so-called integrated S5 function blocks that have a STEP5 access interface have a special function but their implementation is either not written in STEP5 (or MC5) or is protected by a special mechanism. POU's of this kind are generally firmware and can only be "imported as and interface". The implementation portion of this type of POU is empty. These POU's must generally be reprogrammed after being converted.

5. There are also firmware OBs that have no interface but whose code is in 805xx Assembler (as an example) and not in STEP5. This mainly affects the PID regulator listed as OB251 which obtains its parameters and local variables through a separate (data) block that you can select. Neither the PID regulator, the corresponding data block or other POU's that use regulators to access the data block can be converted to IEC. The IEC code that is created for data blocks and other POU's during the conversion is meaningless without the PID regulator. The meaning of the individual program parts can be found in the programming handbook for the CPU.

6. Configuration data blocks (like DB1 [S5-95U], DX0, and DX2) are sometimes used to configure S5 CPUs and other assemblies that were converted into useless IEC POU's. The meaning of much of this type of data can be found in the programming handbook for the CPU. For the rest you must use a S5 programming system that can evaluate the configuration DBs . The configuration affects settings for communication, analog value processing, multiprocessing, etc. Therefore, it is useless to even think about working with these POU's on a non-Siemens SPS.

Once the import is complete, you have to find the errors that are shown and then fix, add to and rewrite the affected spots. These spots are marked with comments like:

(*Warning! Unconvertible STEP5/7 code shown as comment:*)

This is followed by the unconvertible code which is also shown as a comment.

Finally, you must check the addresses. Original Siemens addresses are created during the import. These addresses have the following format:

Bits: Byte-Offset.Bit-Nummer

Non-Bits:Byte-Offset

Also word addresses that follow each other in sequence will overlap (simply due to the fact that the numbers in the addresses are byte offsets). This means that %MW32 and %MW33 have an overlapping byte which is %MB33 (only on a Siemens SPS). On your SPS, %MW32 and %MW33 would not normally have anything to do with each other.

Your PLC may have more hierarchies. For example, non-bits have several interlocking levels ("%MW10.0.0" as WORD). You can either make changes to the addresses to make them compatible with your PLC or you can try to leave them out entirely. Proceed very cautiously! In the original Siemens program, it is quite common that word access and bit or byte access is made in the same memory location. When imported into CoDeSys, accesses of this type will only be correctly compiled for data blocks. In this case, CoDeSys creates WORD variables for the words in the DBs . Then when WORD accesses word x in DB y there are no problems. Attempts to access the left or right byte in word x, a double word or a bit will then be compiled into more complex expressions. This cannot be done with memory locations, inputs or outputs since this can't be done with a standard access method (e.g., word access). If you are working with %MX33.3 and with %MB33 or %MW32 or %MD30 you must go to the effort of converting them manually. The IEC program generated by the CoDeSys import will definitely not work correctly.

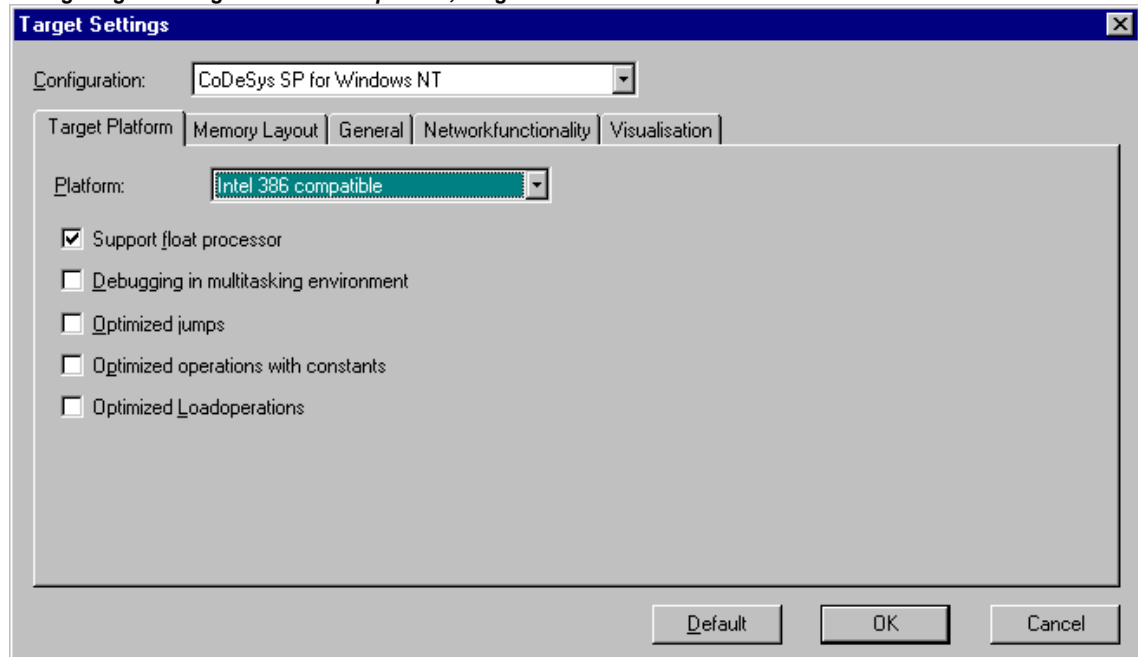
Open a cross reference list containing all inputs, outputs and memory locations to find out which accesses are important. Remove the mixed accessed manually.

Appendix H: Target Settings in Detail

10.29 Settings in Category Target Platform

10.29.1 Target system Intel 386 compatible, Category Target Platform

Dialog Target Settings 'Intel386 compatible', Target Platform



Dialog item	Meaning
Platform	Type of the target-system
Support float processor	if activated: FPU-commands are generated for floating point operations
Debugging in multitasking environment	if activated: additional code is generated, which permits debugging in multitasking environments
Optimized jumps	if activated: optimized conditional jumps after compare operations; faster + less code (especially on 386/486); Lines containing conditions before jumps will be displayed in grey color in flow control mode
Optimized operations with constants	Optimized operations with constants (A = A + 1, A < 500 etc.); Faster + less code (especially on 386/486); Constants will be monitored in grey color in flow control mode
Optimized Loadoperations	No load operations will be executed at multiple access on a variable/constant; Faster + less code

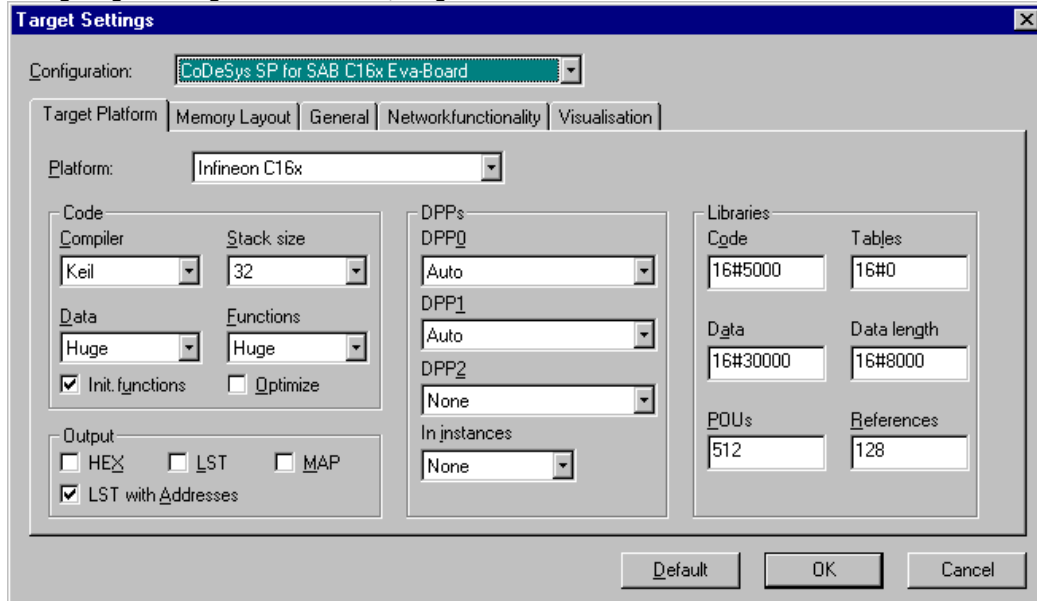
10.29.2 Target system Motorola 68K, Category Target Platform

Dialog Target Settings 'Motorola 68K', Target Platform

Dialog item	Meaning
Platform	Target type
CPU	Variant of the 68k CPU: basic version 68000 or CPU32 and larger
Support Float Processor	if activated: FPU-commands are generated for floating point operations
Use 16 bit jump offsets	if activated: Jumps for evaluating Boolean expressions work with relative 16 bit Offsets (more complex expressions are possible, but larger code) if not activated: 8 bit Offsets are used
allow byte-aligned structures	if activated: only addressing even addresses if not activated: addressing of odd addresses also possible
Reserved Register 1	A2,A4,A5,A6: The indicated address register is reserved and not used If None: it can be used by the code generator
Reserved Register 2	Additional reserved address register. The indicated address register is reserved and not used If "None" it can be used by the code generator
Base register for library data	Register for addressing static data within C libraries (before calling up library functions it is loaded with the address of free memory). If "None" A5 is used as pre-set value.
Output-Mode	Nothing = no output Assembler = During compiling a file "code68k.hex" is created in the compiling directory (Setting under "Project/Options/Directories"). It contains the generated Assembler Code. Disassembler = In addition to 1 the file contains the Disassembler Code

10.29.3 Target system Infineon C16x, Category Target Platform

Dialog Target Settings 'Infineon C16x', Target Platform



Dialog item	Meaning
Platform	Target type
Code / Compiler:	Compiler used during compiling of the target-system and the libraries (on account of C calling conventions)
Code / Stack size	Maximum call depth (nesting)
Code / Data	Memory model for data
Code / Functions	Memory model for code
Init. functions	if activated: Functions contain initialisation code for local variables
Optimize	if activated: Code optimizations for constant array indices
Output HEX-File	if activated: Output of a Hex-Dump of the code
Output BIN-File	if activated: Output of a binary file of the code
Output MAP	if activated=Output of a map-file of the code
Output LST	if activated=Output of a list-file of the code
Output LST , of addresses	if activated=Output of a list of the addresses
DPPs / DPP0..DPP2	Data Page Pointers are set DPP for DPP0, DPP1, DPP2
In Instances	DPP for short addressing of function block Instances
Libraries / Code	Settings for libraries:
Tables	Start addresses for code, tables, data, data length, blocks, references
Data	
Data length	
POUs	
References	

10.29.4 Target systems Intel StrongARM und Power PC, Category Target Platform

The dialog items for these two target systems are identic.

Dialog Target Settings 'PowerPC', Target Platform

Dialog item	Meaning
Platform	Target type
Support float processor	if activated: FPU commands are generated for floating point operations
First parameter Register (integer)	Register where the first Integer Parameter of C-function calls is transmitted (range dependent on the operating system)
Last parameter Register (Integer)	Register where the last Integer Parameter of C-function calls is transmitted (range dependent on the operating system)
Register for return values (Integer)	Register where the Integer Parameters of C-function calls are returned (range dependent on the operating system)
First parameter Register (Float):	Register where the first Float Parameter of C-function calls is transmitted (range dependent on the operating system)
Last parameter Register (Float):	Register where the last Float Parameter of C-function calls is transmitted (range dependent on the operating system)
Register for return value (Float):	Register where the Float Parameters of C-function calls are returned (range dependent on the operating system)
Intel byte order	if activated: Addressing as per Intel address scheme
Max. space on the stack for arguments (Byte)	Size of the stack for arguments (in Bytes). Default: 40

10.29.5 Target system MIPS, Category Target Platform

Dialog Target Settings 'MIPS', Target Platform

Dialog item	Meaning
Platform	Target type
First parameter Register (integer)	Register where the first Integer Parameter of C-function calls is transmitted (range dependent on the operating system)
Last parameter Register (Integer)	Register where the last Integer Parameter of C-function calls is transmitted (range dependent on the operating system)
Register for return values (Integer)	Register where the Integer Parameters of C-function calls are returned (range dependent on the operating system)
Max. space on the stack for arguments (Byte):	Dependent on Operating System: Maximum size (number of bytes) of arguments, which can be handed over on the stack

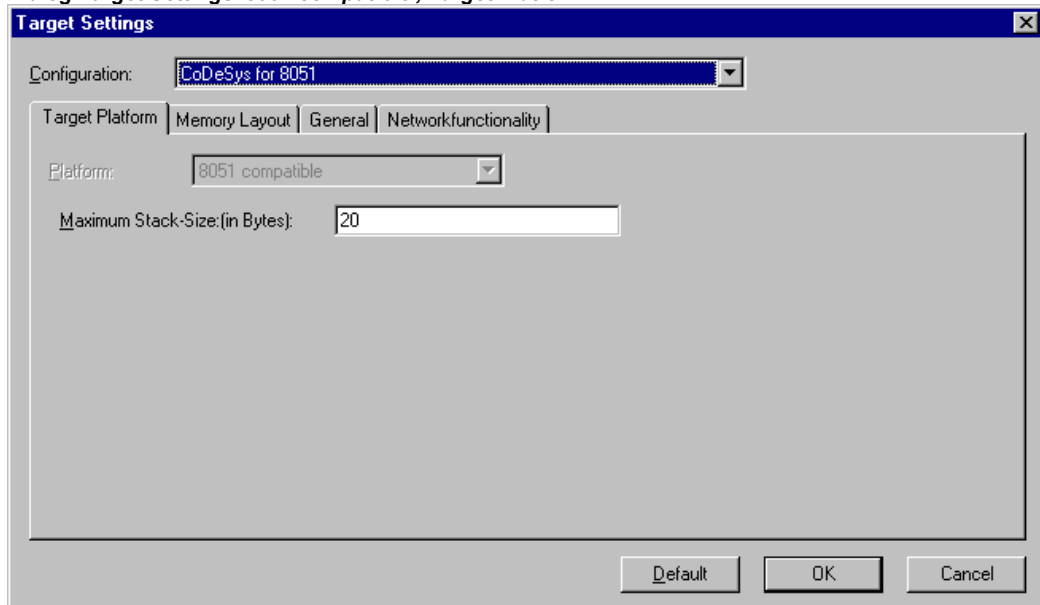
10.29.6 Target system 'Hitachi SH', Category Target Platform

Dialog Target Settings 'Hitachi SH', Target Platform

Dialog item	Meaning
Platform	Target type
First parameter Register (integer)	Register where the first Integer Parameter of C-function calls is transmitted (range depends on the operating system)
Last parameter Register (Integer)	Register where the last Integer Parameter of C-function calls is transmitted (range depends on the operating system)
Register for return values (Integer)	Register where the Integer Parameters of C-function calls are returned (range depends on the operating system)
Max. space on the stack for arguments (Byte):	Dependent on Operating System: Maximum size (number of bytes) of arguments, which can be handed over on the stack
Support float processor	FPU commands are generated for floating point operations
First parameter Register (Float):	Register where the first Float Parameter of C-function calls is transmitted (range depends on the operating system)
Last parameter Register (Float):	Register where the last Float Parameter of C-function calls is transmitted (range depends on the operating system)
Register for return value (Float):	Register where the Float Parameters of C-function calls are returned (range depends on the operating system)
Intel byte order	Intel Byte Adress mode is used

10.29.7 Target system '8051 compatible', Category Target Platform

Dialog Target Settings '8051 compatible', Target Platform



Dialog item	Meaning
Platform	Target type
Maximum Stack-Size:(in Bytes)	Maximum stack size (number of Bytes)

10.29.8 Target System 'TriCore', Categorie Target platform

The target settings for the TriCore target are hard-coded and not modifiable via the target file. If however modifications would be necessary, please contact 3S-Smart Software Solutions GmbH.

Note: The TriCore target system supports native REAL- , not however LREAL-operations.

See here the most important settings:

Einstellung	Bedeutung
Platform=Tricore	Typ des Zielsystems
First Parameter Register (Integer) = 4	Register where the first Integer Parameter of C-function calls is transmitted (range depends on the operating system)
Last Parameter Register (Integer) = 7	Register where the last Integer Parameter of C-function calls is transmitted (range depends on the operating system)
Register for Return values (Integer) = 2	Register where the last Integer Parameter of C-function calls is transmitted (range depends on the operating system)
Others:	- Function call not used for the implementation of breakpoints - Motorola ByteOrder not used - Alignment: 4 Bytes (important for arrays)

10.30 Target Settings for Category Memory Layout

The items described for this tab can be available for each standard target.

Dialog Target Settings 'Intel 386 compatible', Memory Layout

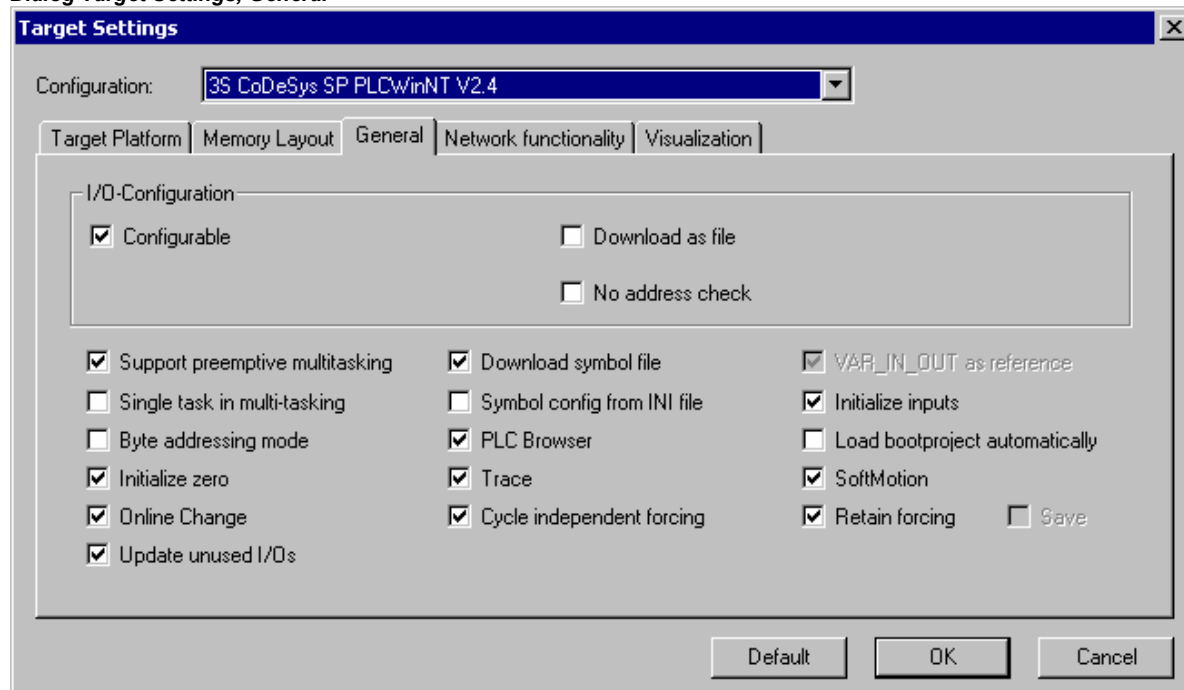
Dialog item	Meaning
Base (Code)	Automatic activated: Code segment is automatically allocated Automatic inactivated: Code segment lies on the given absolute address
Base (Global)	Automatic activated: Data segment (global data) are automatically allocated to the area in question Automatic inactivated: Data segment (global data) lies on the given absolute address
Base (Memory)	Automatic activated: Flags are automatically allocated to the area in question Automatic inactivated: Flag segment lies on the given absolute address
Base (Input)	Automatic activated: Input process image is automatically allocated to the area in question Automatic inactivated: Input process image lies on the given absolute address
Base (Output)	Automatic activated: Output process image is automatically allocated to the area in question Automatic inactivated: Output process image lies on the given absolute address
Base (Retain)	Automatic activated: Retentive data are automatically allocated to the area in question Automatic inactivated: Output process image lies on the given absolute address
Area (Code)	Segment number of the Data segment (Code);
Area (Global)	Segment number of the Data segment (global data);
Area (Memory)	Segment number of the Flag segment;
Area (Input)	Segment number of the Input Process Image
Area (Output)	Segment number of the Output Process Image

Dialog item	Meaning
Area (Retain)	Segment number of the retentive data
Size (Code)	Size of the Code segment
Size pro Segment (Global)	Size of the Data segment
Size (Memory)	Size of the flag segment
Size (Input)	Size of the Input process image
Size (Output)	Size of the Output process image
Size (Retain)	Size of the Segment for retentive data
Total size of data memory	Total memory data size
Own retain segment	if activated: Retentive data are allocated to in separate segment
Total size of data memory	Total size of data memory
Maximum number of global data segments	Maximum number of global data segments
Maximum number of POUs	Maximum number of POUs allowed in a project

10.31 Target Settings in Category General

The items described for this tab can be available for each standard target.

Dialog Target Settings, General



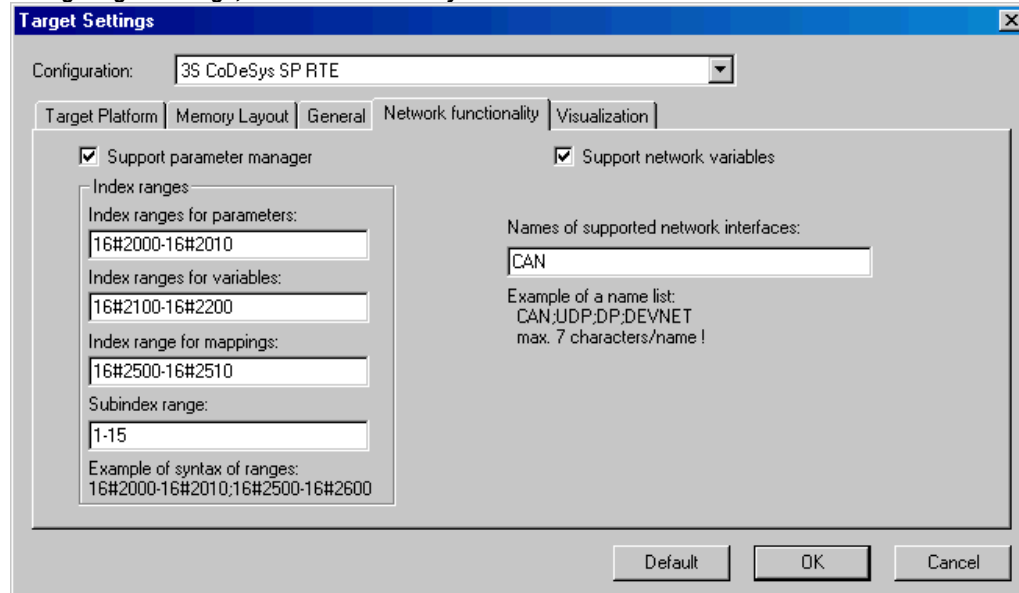
Dialog item	Meaning
Configurable	if activated: Support configurable I/O configurations and load configuration description into the controller
Support CANopen configuration	if activated: Support CANopen configuration and load configuration description into the controller
Support Profibus configuration	if activated: Support Profibus configuration and load configuration description into the controller
Support preemptive multitasking	if activated: Support Task configuration and load task description into the controller
Download as file	if activated: I/O description is downloaded in file format
No address checking	if activated: At compile the IEC addresses are not checked
Online Change	if activated: Online Change functionality
Update unused I/O's	If activated: CoDeSys creates a task for updating inputs and outputs which currently are not used. Thus those I/O's will be monitored in the PLC Configuration.
Singletask in multitasking	not yet implemented
Byte-addressing mode	if activated: byte addressing mode (e.g. var1 AT %QD4 is written in address %QB4)
Initialize zero	if activated: General initialisation with zero
Download Symbol File	if activated: If a symbol file has been created it will be downloaded
Symbol config from INI file	if activated: The parameters for the symbol configuration are not read from the project options dialog, but from the codesys.ini file, resp. from another file which is referenced in the codesys.ini

Dialog item	Meaning
PLC-Browser	if activated: PLC Browser functionality activated
Trace	if activated: Trace functionality activated
VAR_IN_OUT as reference	if activated: At a function call the VAR_IN_OUT variables are called by reference (pointer); therefore no constants can be assigned and no read/write access is possible from outside the function block.
Initialize Inputs	if not activated: For optimizing reasons no init code will be generated for the inputs declared with "AT %IX" (-> undefined values until the 1. bus cycle!)
Automatic boot project load	if activated: A boot project is created automatically after download of a new program and sent to the PLC.
Softmotion	if activated: The SoftMotion functionality is activated, i.e. available in the Resources tab (CNC program list, CAMs)
Retain forcing	if activated: The force list will be kept in the runtime system even at a logout. In this case at logging out the user will get a dialog where he can decide whether the forcing really should be retained. (currently supported by runtime systems CoDeSys SP 32bit full, V2.4, Patch 4 and CoDeSys SP 32bit).
Save	if activated: The runtime system keeps forcing even at a restart. This option is only available if allowed by the target and if option 'Retain forcing' (see above) is activated.
Cycle independent forcing	if activated: Forcing will not only be done at the begin and end of a cycle, but all write accesses during the processing of the program will be deactivated

10.32 Target Settings in Category Networkfunctionality

The items described for this tab can be available for each standard target.

Dialog Target Settings, Networkfunctionality

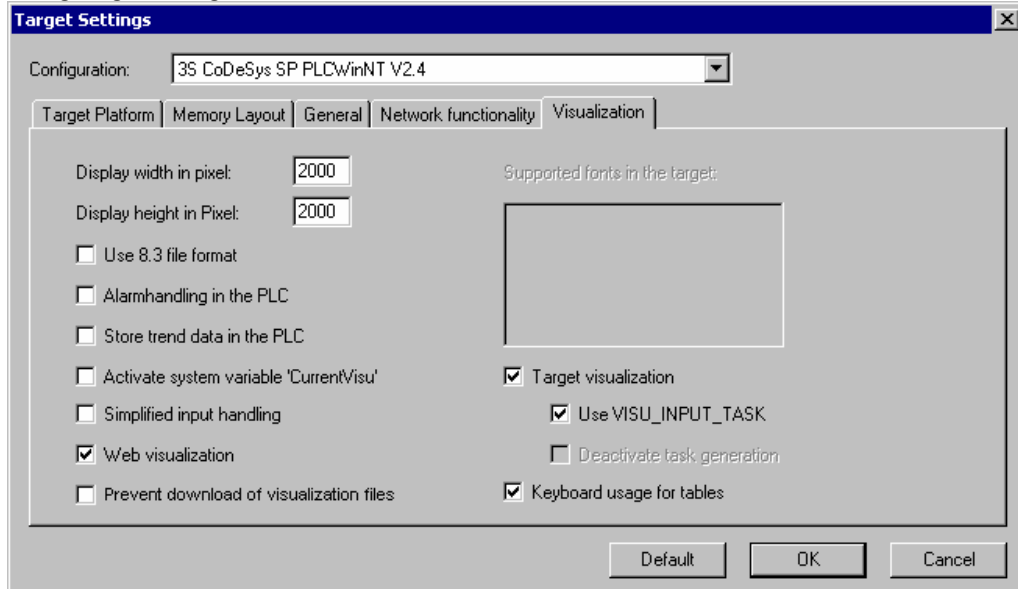


Dialog item	Meaning
Support parameter manager	If activated: the entry 'Parameter-Manager' appears in the Resources tab. Use it to create an Object Dictionary for variables and parameters, which enable the targeted, active data exchange with other controllers
Support network variables	If this option is selected network variables can be used, which enable automatic data exchange in the network
Names of supported networkinterfaces	List of the supported network systems, e.g.: CAN; UDP; DP
Index ranges for parameters	Index Range for Parameter Lists of type 'Parameters' (see Resources, 'Parameter Manager')
Index-ranges for variables	Index Range for Parameter Lists of type 'Variables' (see Resources, 'Parameter Manager')
Index-ranges for Mappings	Index Range for Parameter Lists of type 'Mappings' (see Resources, 'Parameter Manager') Attention: If this range is defined here, the CanDevice will only regard this range for the mapping; that means that if additionally an index range for parameters is defined (see above) that one will not be regarded!
Subindex range	Subindex Range within the above mentioned index ranges for parameter and variable Object Dictionaries (see Resources, 'Parameter Manager')

10.33 Target Settings in Category Visualization

The items described for this tab can be available for each standard target.

Dialog Target Settings, Visualization



Dialog item	Meaning
Display width in pixel Display height in pixel	An area of the given width and height will be displayed in the editor window when editing a visualization. Thus e.g. the size of the screen on which the target visualization will run later, can be regarded when positioning the visualization elements.
Use 8.3 file format	The file names of the bitmaps and language files which are used in the CoDeSys visualization will be shortened to the 8.3-notation format and loaded to the PLC in this format.
Alarmhandling in the PLC	<p>The task ALARM_TASK will be inserted automatically in the task configuration. It will process an implicitly created ST-code evaluating the status of the particular alarms and if applicable executing the associated actions.</p> <p>The ST-code needs auxiliary functions of library SysLibAlarmTrend.lib. This library will be loaded automatically. (Additionally the implicitly needed libraries SysLibSockets.lib, SysLibMem.lib, SysLibTime.lib, SysLibFile.lib are loaded. These libraries must be supported by the target system !)</p> <p>If the option is deactivated and Target visualization is activated, a compile error will be dumped.</p> <p>Hint: The 'Alarm handling in the PLC' can be used even if no Target- or Web-Visualization has been activated. Even then the required ST-code will be generated.</p>
Store trend data in the PLC	<p>The trend handling in the PLC will be activated. The task TREND_TASK will be inserted automatically in the task configuration. It will process an implicitly created ST-code for recording the trend data in a ring buffer and - if option History is activated in the trend element - for storing the values in a file system.</p> <p>The ST-code needs auxiliary functions of library SysLibAlarmTrend.lib. This library will be loaded automatically. (Additionally the implicitly needed libraries SysLibSockets.lib, SysLibMem.lib, SysLibTime.lib, SysLibFile.lib are loaded. These libraries must be supported by the target system !)</p> <p>If the option is deactivated and Target visualization is activated, a compile error will be dumped.</p>

	Hint: 'Store Trend data...' can be used even if no Target- or Web-Visualization has been activated. The required ST-code will be generated.
Activate system variable 'CurrentVisu'	The system variable CurrentVisu can be used for switching between visualizations.
Supported fonts in the target	List of fonts which are supported by the target system.
Simplified input handling	<p>If activated: In online mode the input handling is simplified: <Tab> and <Space> keys are not needed to get from one input field to the next one. The selection is automatically given forward to the next field after having terminated an input by the <Return> key. An input field also can be reached via the arrow or <Tab> keys and then immediately an input can be entered.</p> <p>If not activated: <Tab> and <Space> keys must be used to get to the next input field and to select this field for making possible an input.</p>
Web visualization	if activated: All visualization objects of the project are compiled for the usage as Web visualization objects.
Compression	<p>If activated: The following files for the Web-Visualization, which are to be transferred from Codesys to the Webserver/PLC, will be transferred in a packed format (zip-Format); otherwise in original format:</p> <ul style="list-style-type: none"> - XML visualization files - image files (only *.bmp, because with others no compression effect) - language files (*.xml for dynamic texts, *.tlt, *.vis) <p>The files additionally to the existing file name get the extension „.zip“. The dot in the existing name will be replaced by an underscore (example: „PLC_VISU.xml“ will be renamed to „PLC_VISU_xml.zip“)</p> <p>No compression is done for the Java-archives (minml.jar, webvisu.jar) and the main page webvisu.htm.</p>
Prevent download of visualization files	<p>If activated: When the project is downloaded, all files which are used in the current visualization will not be downloaded to the target system.</p> <p>Visualization files are only downloaded for Target- or Web-Visualization and can be bitmaps, language files and for Web-Visualization also XML description files.</p>
Keyboard usage for tables	If this option is activated, in online mode the keyboard usage of tables in the visualization (CoDeSys HMI, Web-Visualization, Target-Visualization) is possible. Switching off this option will effect that no code is generated for the key functions, which might be reasonable for performance reasons when using Target-Visualization.
Target visualization	if activated: All visualization objects of the project are compiled for the usage as Target visualization objects.
Use VISU_INPUT_TASK	<p>can only be activated if Target-Visualization is activated, see above)</p> <p>If activated and 'Deactivate task generation' (see below) is deactivated, then automatically two tasks will be created for controlling the Target Visualization: VISU_INPUT_TASK controls the processing of the user inputs by means of the implicitly available POU MAINTARGETVISU_INPUT_CODE</p> <p>VISU_TASK controls the repainting of the visualization elements by means of the implicitly available POU MAINTARGETVISU_PAINT_CODE.</p> <p>If the option is deactivated, only VISU_TASK will be created and only POU MAINTARGETVISU_PAINT_CODE is used, which in this case will additionally include the functionality of MAINTARGETVISU_INPUT_CODE.</p>
Deactivate task generation	<p>(can only be activated if Target-Visualization is activated, see above)</p> <p>If the option is activated, the tasks VISU_INPUT_TASK und VISU_TASK (see above at 'Use VISU_INPUT_TASK') will not be generated automatically. Thus the above mentioned POU's, resp. - if option Use VISU_INPUT_TASK' is not activated - only POU MAINTARGETVISU_PAINT_CODE can be called in the application program and, if desired, can be controlled by any task, as defined by the programmer. Referring this see the description on the Target visualization (in the user manual for the CoDeSys Visualization).</p>

Appendix I: Use of Keyboard

10.34 Use of Keyboard

If you would like to run CoDeSys using only the keyboard, you will find it necessary to use a few commands that are not found in the menu.

The function key <F6> allows you to toggle back and forth within the open POU between the Declaration and the Instruction parts.

<Alt>+<F6> allows you to move from an open object to the Object Organizer and from there to the Message window if it is open. If a Search box is open, <Alt>+<F6> allows you to switch from Object Organizer to the Search box and from there back to the object.

Press <Ctrl>+<F6> to move to the next open editor window, press <Ctrl>+<Shift>+<F6> to get to the previous.

Press <Tab> to move through the input fields and buttons in the dialog boxes.

The arrow keys allow you to move through the register cards and objects within the Object Organizer and Library Manager.

All other actions can be performed using the menu commands or with the shortcuts listed after the menu commands. <Shift>+<F10> opens the context menu which contains the commands most frequently used for the selected object or for the active editor.

10.35 Key Combinations

The following is an overview of all key combinations and function keys:

General Functions	
Move between the declaration part and the instruction part of a POU	<F6>
Context Menu	<Shift>+<F10>
Shortcut mode for declarations	<Ctrl>+<Enter>
Move from a message in the Message window back to the original position in the editor	<Enter>
Move to the next open editor window	<Ctrl>+<F6>
Move to the previous open editor window	<Ctrl>+<Shift>+<F6>
Open and close multi-layered variables	<Enter>
Open and close folders	<Enter>
Switch register cards in the Object Organizer and the Library Manager	<Arrow keys>
Move to the next field within a dialog box	<Tab>
Context sensitive Help	<F1>
General Commands	
'File' 'Save'	<Ctrl>+<S>

Appendix I: - Use of Keyboard

'File' 'Print'	<Ctrl>+<P>
'File' 'Exit'	<Alt>+<F4>
'Project' 'Build'	<F11>
'Project' 'Delete Object'	
'Project' 'Add Object'	<Ins>
'Project' 'Rename Object'	<Spacebar>
'Project' 'Open Object'	<Enter>
'Edit' 'Undo'	<Ctrl>+<Z>
'Edit' 'Redo'	<Ctrl>+<Y>
'Edit' 'Cut'	<Ctrl>+<X> or <Shift>+
'Edit' 'Copy'	<Ctrl>+<C>
'Edit' 'Paste'	<Ctrl>+<V>
'Edit' 'Delete'	
'Edit' 'Find next'	<F3>
'Edit' 'Input Assistant'	<F2>
'Edit' 'Auto Declare'	<Shift>+<F2>
'Edit' 'Next Error'	<F4>
'Edit' 'Previous Error'	<Shift>+<F4>
'Online' 'Log-in'	<Alt><F8>
'Online' 'Logout'	<Ctrl>+<F8>
'Online' 'Run'	<F5>
'Online' 'Toggle Breakpoint'	<F9>
'Online' 'Step over'	<F10>
'Online' 'Step in'	<F8>
'Online' 'Single Cycle'	<Ctrl>+<F5>
'Online' 'Write Values'	<Ctrl>+<F7>
'Online' 'Force Values'	<F7>
'Online' 'Release Force'	<Shift>+<F7>
'Online' "Write/Force dialog"	<Ctrl><Shift>+<F7>
'Window' 'Messages'	<Shift>+<Esc>
FBD Editor Commands	
'Insert' 'Network (after)'	<Ctrl>+<T>
'Insert' 'Assignment'	<Ctrl>+<A>
'Insert' 'Jump'	<Ctrl>+<L>

'Insert' 'Return'	<Ctrl>+<R>
'Insert' 'Function Block'	<Ctrl>+
'Insert' 'Input'	<Ctrl>+<U>
'Extras' 'Negate'	<Ctrl>+<N>
'Extras' 'Zoom'	<Alt>+<Enter>
CFC Editor Commands	
'Insert' 'POU'	<Ctrl>+
'Insert' 'Input'	<Ctrl>+<E>
'Insert' 'Output'	<Ctrl>+<A>
'Insert' 'Jump'	<Ctrl>+<J>
'Insert' 'Label'	<Ctrl>+<L>
'Insert' 'Return'	<Ctrl>+<R>
'Insert' 'Comment'	<Ctrl>+<K>
'Insert' 'POU input'	<Ctrl>+<U>
'Extras' 'Negate'	<Ctrl>+<N>
'Extras' 'Set/Reset'	<Ctrl>+<T>
'Extras' 'Connection'	<Ctrl>+<M>
'Extras' 'EN/ENO'	<Ctrl>+<I>
'Extras' 'Zoom'	<Alt>+<Enter>
LD Editor Commands	
'Insert' 'Network (after)'	<Ctrl>+<T>
'Insert' 'Contact'	<Ctrl>+<K>
'Insert' 'Contact (negated)'	<Ctrl>+<G>
'Insert' 'Parallel Contact'	<Ctrl>+<R>
'Insert' 'Parallel contact (negated)'	<Ctrl>+<D>
'Insert' 'Function Block'	<Ctrl>+
'Insert' 'Coil'	<Ctrl>+<L>
'Insert' 'Set coil'	<Ctrl>+<I>
'Insert at blocks' 'Input'	<Ctrl>+<U>
'Insert at blocks' 'Assign'	<Ctrl>+<A>
'Extras' 'Negate'	<Ctrl>+<N>
'Extras' 'Zoom'	<Alt>+<Enter>

SFC Editor Commands	
'Insert' 'Step-Transition (before)'	<Ctrl>+<T>
'Insert' 'Step-Transition (after)'	<Ctrl>+<E>
'Insert' 'Alternative Branch (right)'	<Ctrl>+<A>
'Insert' 'Parallel Branch (right)'	<Ctrl>+<L>
'Insert' 'Jump'	<Ctrl>+<U>
'Extras' 'Zoom Action/Transition'	<Alt>+<Enter>
Move back to the editor from the SFC Overview	<Enter>
Work with the PLC- resp. Task Configuration	
Open and close organization elements	<Enter>
Place an edit control box around the name	<Spacebar>
'Extras' 'Edit Entry'	<Enter>
Working in the Parameter Manager Editor	
Toggle between navigation window and list editor	<F6>
Delete a line in the list editor	<Ctrl>+ <Shift>+
Delete a field in the list editor	

Appendix J: Recommendations on the naming of identifiers

10.36 Naming of identifiers

Identifiers are defined at the declaration of variables (Variable names), user-defined data types and at the creation of POU's (functions, function blocks, programs) and visualizations. You might follow the following recommendations concerning the naming of identifiers in order to make it as unique as possible.

10.37 Identifiers for variables (variable names)

The naming of variables in applications and libraries as far as possible should follow the Hungarian notation:

For each variable a meaningful, short description should be found, the base name. The first letter of each word of a base name should be a capital letter, the others should be small ones (Example: FileSize). If needed additionally an translation file for other languages can be created.

Before the base name, corresponding to the data type of the variable, prefix(es) are added in small letters.

Data type	lower limit	upper limit	Information content	Prefix	Comment
BOOL	FALSE	TRUE	1 Bit	x *	
				b	reserved
BYTE			8 Bit	by	Bit string, not for arithm. operations
WORD			16 Bit	w	Bit string, not for arithm. operations
DWORD			32 Bit	dw	Bit string, not for arithm. operations
LWORD			64 Bit	lw	not for arithm. operations
SINT	-128	127	8 Bit	si	
USINT	0	255	8 Bit	usi	
INT	-32.768	32.767	16 Bit	i	
UINT	0	65.535	16 Bit	ui	
DINT	-2.147.483.648	2.147.483.647	32 Bit	di	
UDINT	0	4.294.967.295	32 Bit	udi	
LINT	-2^{63}	$2^{63} - 1$	64 Bit	li	
ULINT	0	$2^{64} - 1$	64 Bit	uli	
REAL			32 Bit	r	
LREAL			64 Bit	lr	
STRING				s	

TIME				tim	
TIME_OF_DAY				tod	
DATETIME				dt	
DATE				date	
ENUM			16 Bit	e	
POINTER				p	
ARRAY				a	

* pointedly for BOOLEAN variables x is chosen as prefix, in order to differentiate from BYTE and also in order to accommodate the perception of an IEC-programmer (see addressing %IX0.0).

Examples:

```
bySubIndex: BYTE;
sFileName: STRING;
udiCounter: UDINT;
```

In nested declarations the prefixes are attached to each other in the order of the declarations:

Example:

```
pabyTelegramData: POINTER TO ARRAY [0..7] OF BYTE;
```

Function block instances and variables of user-defined data types as a prefix get a shortcut for the FB- resp. data type name (Example: sdo).

Example:

```
cansdoReceivedTelegram: CAN_SDOTelegram;
TYPE CAN_SDOTelegram :      (* prefix: sdo *)
STRUCT
  wIndex:WORD;
  bySubIndex:BYTE;
  byLen:BYTE;
  aby: ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE
```

Local constants (c) start with prefix c and an attached underscore, followed by the type prefix and the variable name.

Example:

```
VAR CONSTANT
  c_uiSyncID: UINT := 16#80;
END_VAR
```

For Global Variables (g) and Global Constants (gc) an additional prefix + underscore is attached to the library prefix:

Examples:

```
VAR_GLOBAL
  CAN_g_iTest: INT;
END_VAR
VAR_GLOBAL CONSTANT
  CAN_gc_dwExample: DWORD;
END_VAR
```

10.38 Identifiers for user-defined data types (DUT)

The name of each structure data type consists of a library prefix (Example: CAN), an underscore and a preferably short expressive description (Example: SDOTelegram) of the structure. The associated prefix for used variables of this structure should follow directly after the colon.

Example:

```
TYPE CAN_SDOTelegram :      (* prefix: sdo *)
STRUCT
  wIndex:WORD;
  bySubIndex:BYTE;
  byLen:BYTE;
  abyData: ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE
```

Enumerations start with the library prefix (Example: CAL), followed by an underscore and the identifier in capital letters..

Regard that in previous versions of CoDeSys ENUM values > 16#7FFF have caused errors, because they did not get converted automatically to INT values. For this reason ENUMs always should be defined with correct INT values.

Example:

```
TYPE CAL_Day : (
  CAL_MONDAY,
  CAL_TUESDAY,
  CAL_WEDNESDAY,
  CAL_THIRSDAY,
  CAL_FRIDAY,
  CAL_SATURDAY,
  CAL_SUNDAY);
```

Declaration:

```
eToday: CAL_Day;
```

10.39 Identifiers for functions, function blocks, programs (POU)

The names of functions, function blocks and programs consist of the library prefix (Example: CAN), an underscore and an expressive short name of the POU (Example: SendTelegram). Like with variables always the first letter of a word of the POU name should be a capital letter, the others should be small letters. It is recommended to compose the name of the POU of a verb and a substantive.

Example:

```
FUNCTION_BLOCK CAN_SendTelegram (* prefix: canst *)
```

In the declaration part a short description of the POU should be provided as a comment. Further on all **inputs** and **outputs** should be provided with **comments**. In case of function blocks the associated prefix for set-up instances should follow directly after the name.

Actions get no prefix; just actions which should be called only internally, i.e. by the POU itself, start with „prv_“.

Each **function** - for the reason of compatibility with previous CoDeSys versions - must have at least one parameter. **External functions** must not use structures as return values.

10.40 Identifiers for Visualizations

Note: Currently you must avoid that a visualization has the same name like another POU in the project. This would lead to problems in case of changes between visualizations.

Appendix K: Compiler Errors and Warnings

If errors are detected during the compilation of the project, messages will be dumped in the message window. Also warnings might be displayed there. <F4> always allows to jump to the next message line in this window, whereby also the concerned POU will be opened. The error and warning messages are preceded by unique numbers. If a message line is selected in the message window, <F1> will open the corresponding online help window.

10.41 Warnings

1100**"Unknown function '<name>' in library."**

An external library is used. Please check, whether all functions, which are defined in the .hex file, are also defined in the .lib file.

1101**"Unresolved symbol '<Symbol>'."**

The code generator expects a POU with the name <Symbol>. It is not defined in the project. Define a function/program with this name.

1102**"Invalid interface for symbol '<Symbol>'."**

The code generator expects a function with the name <Symbol> and exactly one scalar input, or a program with the name <Symbol> and no input or output.

1103**"The constant '<name>' at code address '<address>' overwrites a 16K page boundary!"**

A string constant exceeds the 16K page boundary. The system cannot handle this. It depends on the runtime system whether the problem could be avoided by an entry in the target file. Please contact the PLC manufacturer.

1200**"Task '<name>', call of '<name>' Access variables in the parameter list are not updated"**

Variables, which are only used at a function block call in the task configuration, will not be listed in the cross reference list.

1300**"File not found '<name>'"**

The file, to which the global variable object is pointing, does not exist. Please check the path.

1301**"Analyze-Library not found! Code for analyzation will not be generated."**

The analyze function is used, but the library analyzation.lib is missing. Add the library in the library manager.

1302**"New externally referenced functions inserted. Online Change is therefore no longer possible!"**

Since the last download you have linked a library containing functions which are not yet referenced in the runtime system. For this reason you have to download the complete project.

- 1400**
"Unknown Pragma '<Name>' is ignored!"
 This pragma is not supported by the compiler. See keyword 'pragma' for supported directives.
- 1401**
"The struct '<name>' does not contain any elements."
 The structure does not contain any elements, but variables of this type allocate 1 Byte of memory.
- 1410**
""RETAIN' and 'PERSISTENT' do not have any effect in functions"
 Remanent variables which are defined locally in functions are handled like normal local variables.
- 1411**
"Variable '<name>' in the variable configuration isn't updated in any task"
 The top level instance of the variable is not referenced by a call in any task. Thus it will not be copied from the process image.
Example:
 Variable Configuration:

```
VAR_CONFIG
  plc_prg.aprg.ainst.in AT %IB0 : INT;
END_VAR
plc_prg:
  index := INDEXOF(aprg);
```

 The program aprg is referenced but not called. Thus plc_prg.aprg.ainst.in never will get the actual value of %IB0.
- 1412**
"Unexpected token '<Name>' in pragma {pragma name}"
 You are using a pragma which is not written correctly resp. which cannot be used at this location. See keyword 'pragma' in the CoDeSys Online Help or Users Guide for getting help for a correction.
- 1413**
""<Name>' is not a valid key for list '<Name>'. The key will be ignored"
 In the pragma a nonexistent parameter list is specified. Check the list name resp. have a look in the Parameter Manager for the currently available lists.
- 1414**
""Too many component definitions in pragma '<Name>""
 The Pragma contains more definitions (in square brackets) than there are elements in the corresponding array, functionblock or structure.
- 1415**
""<Name>' (<Number>): The literal '<Number>' is assigned to more than one enumeration"
 In the declaration of enumeration <Name> the same number is assigned to more than one enumeration components (e.g. TYPE aenum (a:=1, b:=1); END_TYPE).
- 1500**
"Expression contains no assignment. No code was generated."
 The result of this expression is not used. For this reason there is no code generated for the whole expression.

1501

"String constant passed as 'VAR_IN_OUT': '<Name>' must not be overwritten!"

The constant may not be written within the POU, because there no size check is possible.

1502

"Variable '<Name>' has the same name as a POU. The POU will not be called!"

A variable is used, which has the same name like a POU.

Example:

```
PROGRAM a
  . . .
  VAR_GLOBAL
  a: INT;
  END_VAR
...
a; (* Not POU a is called but variable a is loaded. *)
```

1503

"The POU '<name>' has no outputs. Box result is set to 'TRUE'."

The Output pin of a POU which has no outputs, is connected in FBD or KOP. The assignment automatically gets the value TRUE.

1504

"'<name>' ('<number>'): Statement may not be executed due to the evaluation of the logical expression"

Eventually not all branches of the logic expression will be executed.

Example:

```
IF a AND funct(TRUE) THEN ....
If a has is FALSE then funct will not be called.
```

1505

"Side effect in '<Name>'! Branch is probably not executed !"

The first input of the POU is FALSE, for this reason the side branch, which may come in at the second input, will not be executed.

1506

"Variable '<name>' has the same name as a local action. The action will not be called!"

Rename the variable or the action.

1507

"Instance '<name>' has the same name as a function. The instance will not be called."

You call in ST an instance which has the same name like a function. The function will be called ! Use different names.

1509

"'<name>' ('<number>'): Functions to be registered as callbacks have to start with 'Callback'."

You are using a callback function, the name of which does not start with "callback". This can cause unexpected effects on RISC resp. Motorola 68K controllers!

- 1510**
"Operand to be rotated has no explicit type. Please use a typed literal, like 'DWORD#1'."
 For the constant used in the ROL resp. ROR operation there must be a data type definition. For information on typed constants see [HERE](#).
- 1511**
"Operand to be shifted has no explicit type. Please use a typed literal, like 'DWORD#1'."
 For the constant used in the SHL resp. SHR operation there must be a data type definition. For information on typed constants see [HERE](#).
- 1550**
"Multiple calls of the POU '<Name>' in one network may lead to undesired side effects"
 Check, whether the multiple call of this POU is really necessary. By a multiple call unwanted value overstrikes may occur.
- 1600**
"Open DB unclear (generated code may be erroneous)."
 The original Siemens program does not tell, which POU is opened.
- 1700**
"Input not connected."
 An input box is used in CFC which has no assignment. For this no code will be generated.
- 1750**
"Step '<Name>': the minimal time is greater than the maximal time!"
 Open dialog 'Step attributes' for this step and correct the time definitions.
- 1800**
"<name>(element #<element number>): Invalid watchexpression '<name>'"
 The visualization element contains an expression which cannot be monitored. Check variable name and placeholder replacements.
- 1801**
"<name> (number): No Input on Expression '<name>' possible"
 In the configuration of the visualization object at field input a composed expression is used. Replace this by a single variable.
- 1802**
"<Visualization object>(Element number): Bitmap '<name>' was not found"
 Make sure, that an external bitmap-file is available in that path which is defined in the visualization configuration dialog.
- 1803**
"<name>'(<number>)': 'The print action would not supported for web- and target visualization"
 A print action is assigned to an alarm configured in the visualization. This will not be regarded in the Web- or Target-Visualization.

1804

""<name>'(<number>): The font '<name>' is not supported by the target."

In the visualization you are using a font, which is not supported by the target system. See in the target settings, category 'Visualization' for the supported fonts.

1807

""<name> (<number>): No message window for alarms for target visualization"

Regard that action "message" is not supported for the Target-Visualization!

1808

""<name>'(<number>): A polygon consists of to many points for target visualization. In case of a meter element, please open the configuration once."

Per default maximum 512 points are allowed, target-specifically another maximum number might be defined. By opening the configuration dialog the element will be optimized to the allowed number of points.

1809

""<name> (<number>): Invalid visualization as zoom target"

This visualization could not be found. Please check whether the visualization name is not correct or the visualization object is not available at all.

1850

"Input variable at %IB<number> is used in task '<name>' but updated in another task"

Please check which tasks are using this variable and whether the current programming is not causing undesirable effects. The update of the variable value usually is done in the task with the highest priority.

1851

"Output variable at %IQ<number> is used in task '<name>' but updated in another task"

Please check which tasks are using this variable and whether the current programming is not causing undesirable effects. The update of the variable value usually is done in the task with the highest priority.

1852

"CanOpenMaster might not be called cyclically in event task '<name>'! Set modul parameter UpdateTask!"

Currently the CanOpen Master is called by the named event task. If you want to get it called cyclically, specify an appropriate task via parameter UpdateTask in the PLC Configuration in dialog 'Module parameters'.

1853

"A PDO (index: '<number>') might not be updated cyclically in event task '<name>'"

Currently the named PDO is controlled via the named event task. But if you want to get it called cyclically, you must assign an appropriate task to the PDO by shifting IO-references to this task.

1900

"POU '<name>' (main routine) is not available in the library"

The Start-POU (e.g. PLC_PRG) will not be available, when the project is used as library.

1901

"Access Variables and Variable Configurations are not saved in a library!"

Access variables and variable configuration are not stored in the library.

- 1902**
""<Name>': is no Library for the current machine type!"
 The .obj file of the lib was generated for another device.
- 1903**
""<Name>: is no valid Library"
 The file does not have the format requested for the actual target.
- 1904**
"The constant '<Name>' hides a constant of the same name in a library"
 In your project you have defined a constant which has the same name like one which is defined in a linked library. The library variable will be overwritten !
- 1970**
"Parameter manager: List '<Name>' , Column '<Name>', Value '<Name>' could not be imported!"
 Check the Import-file *.prm for entries which do not match the current configuration (standard values resp. XML-description file) of the Parameter Manager.
- 1980**
Global network variables '<Name>' '<Name>': simultaneous reading and writing may result in loss of data!"
 In the configuration of the network variables list (Select list in the Resources tab and open dialog 'Global variables list' via command 'Properties' in the context menu) options 'Read' and 'Write' are activated. Regard that this might result in data losses during communication.
- 1990**
"No 'VAR_CONFIG' for '<name>'"
 For this variable there is no address configuration available in the Variable_Configuration (VAR_CONFIG). Open window Variable_Configuration in the Resources tab and there insert the appropriate configuration (Command 'Insert 'All instance paths').
- 2500**
"Task '<task name>': no cycle time specified for cyclic task"
 In the Task configuration a cyclic task has been created, for which no cycle time has been defined. Enter an appropriate time span in dialog 'Taskattributes' at "Interval".
- 4710**
""<name>' ('<number>'): A required expression has not been configured"
 In the configuration of the slider element in category 'Variables' there is not entered a valid value for each of the fields 'Min.Value', 'Slider' and 'Max. Value'.
- 4711**
""<name>' ('<number>'): The slider element is not supported by your SysLibTargetVisu."
 Install and use the current version of the library SysLibTargetVisu.library.
- 4714**
""<name>' ('<number>'): The type of the selection variable has to be a signed type compatible to INT""

Check and appropriately correct the configuration of the table element in category "Selection".

4715

"The current target does not support formatted display of time and date values"

Option "Formatted time display" is activated in the settings of a visualization. However this is not supported by the currently used target system.

4716

"The current target does not support simplified input handling"

In the target settings, in category 'Visualization' option 'Simplified input handling' is activated. Maybe it is not supported by the current target system, because an old version of the library SysLibTargetvisu.lib is used.

10.42 Errors

3100

"Code too large. Maximum size: '<number>' Byte (<number>K)"

The maximum program size is exceeded. Reduce project size.

3101

"Total data too large. Maximum size: '<number>' Byte (<number>K)"

Memory is exceeded. Reduce data usage of the application.

3110

"Error in Library '<Name>'."

The .hex file is not in INTEL Hex format.

3111

"Library '<Name>' is too large. Maximum size: 64K"

The .hex file exceeds the set maximum size.

3112

"Nonrelocatable instruction in library."

The .hex file contains a instruction which is not relocatable. The library code cannot be linked.

3113

"Library code overwrites function tables."

The ranges for code and function tables are overlapping.

3114

"Library uses more than one segment."

The tables and the code in the .hex file use more than one segment.

- 3115**
"Unable to assign constant to VAR_IN_OUT. Incompatible data types."
 The internal pointer format for string constants cannot get converted to the internal pointer format of VAR_IN_OUT, because the data are set "near" but the string constants are set "huge" or "far". If possible change these target settings.
- 3116**
"Function tables overwrite library code or a segment boundary."
 Code 166x: The external library cannot be used with the current target settings. These must be adapted resp. the library must be rebuilt with appropriate settings.
- 3117**
"<Name> (<Zahl>): Expression too complex. No more registers available"
 The named expression is too complex to be handled by the available registers. Please try to reduce the expression by using interim variables.
- 3120**
"Current code-segment exceeds 64K."
 The currently generated code is bigger than 64K. Eventually too much initializing code is created.
- 3121**
"POU too large."
 A POU may not exceed the size of 64K.
- 3122**
"Initialisation too large. Maximum size: 64K"
 The initialization code for a function or a structure POU may not exceed 64K.
- 3123**
"Data segment too large: segment '<Number>%s', size <size> bytes (maximum <number> bytes)"
 Please contact your manufacturer.
- 3124**
"String-Konstante zu groß: <Anzahl> Zeichen (Maximum 253 Zeichen)"
 The given constant must be reduced in number of characters.
- 3130**
"User-Stack too small: '<number>' DWORD needed, '<number>' DWORD available."
 The nesting depth of the POU calls is too big. Enter a higher stack size in the target settings or compile build project without option ,Debug' (can be set in dialog 'Project' 'Options' 'Build').
- 3131**
"User-Stack too small: '<number>' WORD needed, '<number>' WORD available."
 Please contact the PLC manufacturer.
- 3132**
"System-Stack too small: '<number>' WORD needed, '<number>' WORD available."
 Please contact the PLC manufacturer.

3150

"Parameter <number> of function '<name>': Cannot pass the result of a IEC-function as string parameter to a C-function."

Use a intermediate variable, to which the result of the IEC function is assigned.

3160

"Can't open library file '<name>'."

A library <name> is included in the library manager for this project, but the library file does not exist at the given path.

3161

"Library '<name>' contains no code segment"

An .obj file of a library at least must contain one C function. Insert a dummy function in the .obj file, which is not defined in the .lib file.

3162

"Could not resolve reference in Library '<name>'(Symbol '<name>', Class '<name>', Type '<name>')"

The .obj file contains a not resolvable reference to another symbol. Please check-the settings of the C-Compiler.

3163

"Unknown reference type in Library '<name>' (Symbol '<name>', Class '<name>', Type '<name>')"

The .obj file contains a reference type, which is not resolvable by the code generator. Please check-the settings of the C-Compiler.

3200

"<name>: Boolean expression to complex"

The temporary memory of the target system is insufficient for the size of the expression. Divide up the expression into several partial expressions thereby using assignments to intermediate variables.

3201

"<name> (<network>): A network must not result in more than 512 bytes of code"

Internal jumps can not be resolved. Activate option "Use 16 bit jump offsets" in the 68k target settings.

3202

"Stack overrun with nested string/array/structure function calls"

A nested function call CONCAT(x, f(i)) is used. This can lead to data loss. Divide up the call into two expressions.

3203

"Expression too complex (too many used adress registers)."

Divide up the assignment in several expressions.

3204

"A jump exceeds 32k Bytes"

Jump distances may not be bigger than 32767 bytes.

- 3205**
"Internal Error: Too many constant strings"
In a POU there at the most 3000 string constants may be used.
- 3206**
"Function block data exceeds maximal size"
A function block may produce maximum 32767 Bytes of code.
- 3207**
"Array optimization"
The optimization of the array accesses failed because during index calculation a function has been called.
- 3208**
"Conversion not implemented yet"
A conversion function is used, which is not implemented for the actual code generator.
- 3209**
"Operator not implemented"
A operator is used, which is not implemented for this data type and the actual code generator.
MIN(string1,string2).
- 3210**
"Function '<Name>' not found"
A function is called, which is not available in the project.
- 3211**
"Max string usage exceeded"
A variable of type string can be used in one expression 10 times at the most.
- 3212**
"Wrong library order at POU <POU name>"
The order of libraries for this POU does not match with that in the cslib.hex file. Correct the order accordingly. (only for 68K targets, if the checking option is activated in the target file.)
- 3250**
"Real not supported for 8 Bit Controller"
The target is currently not supported.
- 3251**
"date of day types are not supported for 8 Bit Controller"
The target is currently not supported.
- 3252**
"size of stack exceeds <number> bytes"
The target is currently not supported.
- 3253**
"Could not find hex file: '<Name>' "
The target is currently not supported.

3254

"Call to external library function could not be resolved."

The target is currently not supported.

3255

"Pointers are not supported for 8 bit controllers."

Avoid using pointers in your program to get it running on the 8 bit system.

3260

"Function ,<name>' has too many arguments: Increase the size of the argument stack in the target settings."

If possible, modify the size of the stack in dialog ,Target Platform' in the Target Settings (Default: 40), see chapter 10.29.4. If the setting is not editable in CoDeSys, please contact the PLC manufacturer.

3400

"An error occurred during import of Access variables"

The .exp file contains an incorrect access variables section.

3401

"An error occurred during import of variable configuration"

The .exp file contains an incorrect configuration variables section.

3402

"An error occurred during import of global variables"

The .exp file contains an incorrect global variables section.

3403

"Could not import <name>"

The section for object <name> in the .exp file is not correct.

3404

"An error occurred during import of task configuration"

The section for the task configuration the .exp file is not correct.

3405

"An error occurred during import of PLC configuration"

The section for the PLC configuration in the .exp file is not correct.

3406

"Two steps with the name '<name>'. Second step not imported."

The section for the SFC POU in the .exp file contains two steps with equal names. Rename one of the steps in the export file.

3407

"Predecessor step '<name>' not found"

The step <name> is missing in the .exp file.

3408

"Successor step '<name>' not found"

The step <name> is missing in the .exp file.

- 3409**
"No succeeding transition for step '<name>' "
In the .exp file a transition is missing, which requires step <name> as preceding step.
- 3410**
"No succeeding step for transition '<name>'"
In the .exp file a step is missing which requires the transition <name> as preceding condition.
- 3411**
"Step '<name>' not reachable from initial step"
In the .exp file the connection between step <name> and the initial step is missing.
- 3412**
"Macro '<name>' not imported"
Check the export file.
- 3413**
"Error during import of the CAMs."
You have imported an export file (*.exp) which contains erroneous information on a CAM. Check the export file.
- 3414**
"Error during import of the CNC program list"
You have imported an export file (*.exp) which contains erroneous information on a CNC program. Check the export file.
- 3415**
Error during import of the Alarm configuration"
You have imported an export file (*.exp) which contains erroneous information on the Alarm Configuration. Check the export file.
- 3450**
"PDO'<PDO-name>': Missing COB-Id!"
Click on the button ‚Properties‘ in the PLC configuration dialog for the module and enter a COB ID for the PDO <PDO Name>.
- 3451**
"Error during load: EDS-File '<name>' could not be found, but is referenced in hardware configuration!"
Eventually the device file needed for the CAN configuration is not in the correct directory. Check the directory setting for configuration files in ‚Project‘ ‚Options‘ ‚Directories‘.
- 3452**
"The module '<name>' couldn't be created!"
The device file for module <name> does not fit to the current configuration. Eventually it has been modified since the configuration has been set up in CoDeSys or it is corrupted.
- 3453**
"The channel '<name>' couldn't be created!"
The device file for channel <name> does not fit to the current configuration. Eventually it has been modified since the configuration has been set up in CoDeSys or it is corrupted.

3454

"The address '<name>' points to an used memory!"

Option 'Check for overlapping addresses' is activated in the dialog ,Settings' of the PLC configuration and an overlap has been detected. Regard, that the area check is based on the size which results of the data types of the modules, not on the size which is given by the entry ,size' in the configuration file.

3455

"Error during load: GSD-File '<name>' could not be found, but is referenced in hardware configuration!"

Eventually the device file required by the Profibus configuration is not in the correct directory. Check the directory setting for configuration files in ,Project' 'Options' 'Directories'.

3456

"The profibus device '<name>' couldn't be created!"

The device file for module <name> does not fit to the current configuration. Eventually it has been modified since the configuration has been set up in CoDeSys or it is corrupted.

3457

"Error in module description!"

Please check the device file of this module.

3458

"The PLC-Configuration couldn't be created! Check the configuration files."

Check if all required configuration and device files are available in the correct path (see defined compile directory in 'Project' 'Options' /Directories)

3460

3S_CanDrv.lib has the wrong version.

Make sure, that the 3S_CanDrv.lib which is included in the project is up to date.

3461

"3S_CanOpenMaster.lib has the wrong version."

Make sure, that the 3S_CanOpenMaster.lib which is included in the project is up to date.

3462

"3S_CanOpenDevice.lib has the wrong version."

Make sure, that the 3S_CanOpenDevice.lib which is included in the project is up to date.

3463

"3S_CanOpenManager.lib has the wrong version."

Make sure, that the 3S_CanOpenManager.lib which is included in the project is up to date.

3464

"3S_CanNetVar.lib has the wrong version."

Make sure, that the 3S_CanNetVar.lib which is included in the project, is up to date.

3465

"CanDevice: Sub indices have to be numerated sequentially"

In parameter lists used by the CanDevice the subindices must be numbered sequentially and without interruption. Check the corresponding list in the Parameter Manager.

- 3466**
"CAN network variables: No CAN controller found in the PLC configuration"
 There are network variables configured for a CAN network (Resources, Global Variables), but in the PLC Configuration there is no CAN Controller available.
- 3468**
"CanDevice: Update task not available in the task configuration."
 The update task (used for calling the CANdevice), which is defined in the Base Settings dialog of the CANdevice in the PLC Configuration, must be configured in the Task Configuration of the project.
- 3469**
"The CanOpenMaster can not be called. Please assign a task manually."
 Assign a task, which should call the Master, via parameter UpdateTask in the Module parameters dialog in the PLC Configuration.
- 3470**
"Invalid name in parameter UpdateTask"
 Open the CanMasters Module parameter dialog in the PLC Configuration. Check parameter UpdateTask. The specified task must be available in the project. If you cannot set an appropriate task here, the device file must be checked for the corresponding value definitions for UpdateTask.
- 3500**
"No 'VAR_CONFIG' for '<Name>'"
 Insert a declaration for this variable in the global variable list which contains the 'Variable_Configuration'.
- 3501**
"No address in 'VAR_CONFIG' for '<name>'."
 Assign an address to this variable in the global variable list which contains the 'Variable_Configuration'.
- 3502**
"Wrong data type for '<name>' in 'VAR_CONFIG'"
 In the global variables list which contains the 'Variable_Configuration' the variable is declared with a different data type than in the POU.
- 3503**
"Wrong data type for '<name>' in 'VAR_CONFIG'"
 In the global variables list which contains the 'Variable_Configuration' the variable is declared with a different address than in the POU.
- 3504**
"Initial values are not supported for 'VAR_CONFIG'"
 A variable of the 'Variable_Configuration' is declared with address and initial value. But an initial value can only be defined for input variables without address assignment.
- 3505**
"'<name>' is no valid instance path"
 The Variable_Configuration contains a nonexisting variable.

3506

"Access path expected"

In the global variable list for Access Variables the access path for a variable is not correct. Correct: <Identifier>:<Access path>:<Type> <Access mode>.

3507

"No address specification for 'VAR_ACCESS'-variables"

The global variable list for Access Variables contains an address assignment for a variable. This is not allowed.

Valid variable definition: <Identifier>:<Access path>:<Type> <Access mode>

3550

"Duplicate definition of identifier '<name>'"

There are two tasks are defined with an identic same name. Rename one of them.

3551

"The task '<name>' must contain at least one program call"

Insert a program call or delete task.

3552

"Event variable '<name>' in task '<name>' not defined"

There is an event variable set in the 'Single' field of the task properties dialog which is not declared globally in the project. Use another variable or define the variable globally.

3553

"Event variable '<name>' in task '<name>' must be of type 'BOOL'"

Use a variable of type BOOL as event variable in the 'Single' field of the task properties dialog.

3554

"Task entry '<name>' must be a program or global function block instance"

In the field 'Program call' a function or a not defined POU is entered. Enter a valid program name.

3555

"The task entry '<name>' contains invalid parameters"

In the field 'Append program call' there are parameters used which do not comply with the declaration of the program POU.

3556

"Tasks are not supported by the currently selected target"

The currently defined task configuration cannot be used for the currently set target system. Change target or modify the task configuration correspondingly.

3557

"Maximum number of Tasks ('<number>') exceeded"

The currently defined number of tasks exceeds the maximum number allowed for the currently set target system. Change target or modify the task configuration correspondingly. Attention: Do not edit the XML description file of the task configuration!

3558

"Priority of task '<name>' is out of valid range between '<lower limit>' and '<upper limit>'"

The currently defined priority for the task is not valid for the currently set target system. Change target or modify the task configuration correspondingly.

- 3559**
"Task '<name>': Interval-Tasks are not supported by the current target"
The current task configuration contains an interval task. This is not allowed by the currently set target system. Change target or modify the task configuration correspondingly.
- 3560**
"Task '<name>': free wheeling tasks are not supported by the current target"
The current task configuration contains an free wheeling task. This is not allowed by the currently set target system. Change target or modify the task configuration correspondingly.
- 3561**
"Task '<name>': event tasks are not supported by the current target"
The current task configuration contains event tasks which are not supported by the currently set target system. Change target or modify the task configuration correspondingly.
- 3562**
"Task '<name>': external event tasks are not supported by the current target"
The current task configuration contains external event tasks which are not supported by the currently set target system. Change target or modify the task configuration correspondingly.
- 3563**
"The interval of task '<name>' is out of valid range between '<lower limit>' and '<upper limit>'"
Change the interval value in the configuration dialog for the task.
- 3564**
"The external event '<name>' of task '<name>' is not supported by the current target"
The currently set target system does not support the external event which is defined in the task configuration for this task. Change target or modify the task configuration correspondingly.
- 3565**
"Maximum number of event tasks ('<number>') exceeded"
The currently set target system does not allow as many event tasks as are defined at the moment. Change target or modify the task configuration correspondingly.
- 3566**
"Maximum number of interval tasks ('<number>') exceeded"
The currently set target system does not allow as many interval tasks as defined at the moment. Change target or modify the configuration correspondingly.
- 3567**
"Maximum number of free wheeling tasks ('<number>') exceeded"
The currently set target system does not allow as many free wheeling tasks as defined at the moment. Change target or modify the configuration correspondingly.
- 3568**
"Maximum number of external interval tasks ('<number>') exceeded"
The currently set target system does not allow as many external interval tasks as defined at the moment. Change target or modify the configuration correspondingly.

3569

"POU '<name>' for system event '<name>' not defined"

The POU which should be called by the named system event, as defined in the task configuration, is not available in the project. Modify the task configuration correspondingly or make sure that the POU is available in the project.

3570

"The tasks '<name>' and '<name>' share the same priority"

Modify the task configuration so that each task has a different priority.

3571

"The library 'SysLibCallback' is not included in the project! System events can not be generated."

In order to create event tasks, the SysLibCallback.lib is needed. Link this library to the project in the library manager or modify the task configuration (task attributes) in that way that there is no task triggered by an event.

3575

"Task '<name>': the cycle time has to be a multiple of <number> µs."

Correct the cycle time accordingly in the Taskattributes dialog for this task. The target system defines a base time and prescribes that the cycle time must be equal to or be a multiple of this base time.

3600

"Implicit variables not found!"

Use command ,Rebuild all'. If nevertheless you get the error message again please contact the PLC manufacturer.

3601

"<name> is a reserved variable name"

The given variable is declared in the project, although it is reserved for the codegenerator. Rename the variable.

3610

" '<Name>' not supported"

The given feature is not supported by the current version of the programming system.

3611

"The given compile directory '<name>' is invalid"

There is an invalid directory given in the ,Project' ,Options' ,Directories' for the Compile files.

3612

"Maximum number of POUs (<number>) exceeded! Compile is aborted."

Too many POUs and data types are used in the project. Modify the maximum number of POUs in the Target Settings / Memory Layout.

3613

"Build canceled"

The compile process was cancelled by the user.

- 3614**
"Project must contain a POU named '<name>' (main routine) or a taskconfiguration"
 Create an init POU of type Program (e.g. PLC_PRG) or set up a task configuration.
- 3615**
"<Name> (main routine) must be of type program"
 A init POU (e.g. PLC_PRG) is used in the project which is not of type Program.
- 3616**
"Programs musn't be implemented in external libraries"
 The project which should be saved as an external library contains a program. This will not be available, when the library will be used.
- 3617**
"Out of memory"
 Increase the virtual memory capacity of your computer.
- 3618**
"BitAccess not supported in current code generator!"
 The code generator for the currently set target system does not support bit access on variables.
- 3619**
"Object file '<name>' and library '<name>' have different versions!"
 Make sure that for the library there are available matching versions of *.lib and *.obj resp. *.hex files. These files must have the very same time stamp.
- 3620**
"The POU '<name>' must not be present inside a library"
 You want to save the project as a library of version 2.1. In this version a library may not contain a PLC_PRG object. Use a different POU name.
- 3621**
"Cannot write compile file '<name>'"
 Probably in the path which is specified for the compile file there is already a file of the same name, which is "read only". Remove that file resp. change the access rights.
- 3622**
"The symbol file '<name>' could not be created"
 Probably in the path which is specified for the symbol file (usually project directory) there is already a file of the same name, which is "read only". Remove that file resp. change the access rights.
- 3623**
"Cannot write boot project file '<name>'"
 Probably in the path which is specified for the symbol file (target specific) there is already a file of the same name, which is "read only". Remove that file resp. change the access rights.
- 3624**
"Target setting <targetsetting1>=<set value> not compatible with <targetsetting2>=<set value>"
 Check and correct these settings in the Targetsettings dialogs (Resources tab). If the settings are not visible resp. editable there, please contact the PLC Manufacturer.

3700

"POU with name '<name>' is already in library '<name>'"

A POU name is used in the project, which is already used for a library POU. Rename the POU.

3701

"Name used in interface is not identical with POU Name"

Use command **'Project' 'Rename object'** to rename the POU in the object organizer, or change the name of the POU in the declaration window. There the POU name has to be placed next to one of the keywords PROGRAM, FUNCTION or FUNCTIONBLOCK.

3702

"Overflow of identifier list"

Maximum 100 identifiers can be entered in one variable declaration.

3703

"Duplicate definition of identifier '<Name>'"

Take care that there is only one identifier with the given name in the declaration part of the POU.

3704

"data recursion: "<POU 0> -> <POU 1> -> .. -> <POU 0>""

An instance of a function block is used, which calls itself.

3705

"<Name>: VAR_IN_OUT in Top-Level-POU not allowed, if there is no Task-Configuration"

Create a task configuration or make sure that there are no VAR_IN_OUT variables used in PLC_PRG.

3720

"Address expected after 'AT'"

Add a valid address after the keyword AT or modify the keyword.

3721

"Only 'VAR' and 'VAR_GLOBAL' can be located to addresses"

Put the declaration to a VAR or VAR_GLOBAL declaration area.

3722

"Only 'BOOL' variables allowed on bit addresses"

Modify the address or modify the type of the variable to which the address is assigned.

3726

"Constants can not be laid on direct addresses"

Modify the address assignment correspondingly.

3727

"No array declaration allowed on this address"

Modify the address assignment correspondingly.

- 3728**
"Invalid address: '<address>'"
This address is not supported by the PLC configuration. Check PLC configuration resp. modify address.
- 3729**
"Invalid type '<name>' at address: '<Name>' "
The type of this variable cannot be placed on the given address. Example: For a target system working with 'alignment 2' the following declaration is not valid: var1 AT %IB1:WORD; This error message also might indicate that an array is assigned to the address of a direct variable, which is not allowed.
- 3740**
"Invalid type: '<Name>' "
An invalid data type is used in a variable declaration.
- 3741**
"Expecting type specification"
A keyword or an operator is used instead of a valid type identifier.
- 3742**
"Enumeration value expected"
In the definition of the enumeration type an identifier is missing after the opening bracket or after a comma between the brackets.
- 3743**
"Integer number expected"
Enumerations can only be initialized with numbers of type INT.
- 3744**
"Enum constant '<name>' already defined"
Check if you have followed the following rules for the definition of enumeration values:
Within one enum definition all values have to be unique.
Within all global enum definitions all values have to be unique.
Within all local enum definitions all values have to be unique.
- 3745**
"Subranges are only allowed on Integers!"
Subrange types can only be defined resting on integer data types.
- 3746**
"Subrange '<name>' is not compatible with Type '<name>'"
One of the limits set for the range of the subrange type is out of the range which is valid for the base type.
- 3747**
"unknown string length: '<name>'"
There is a not valid constant used for the definition of the string length.

3748

"More than three dimensions are not allowed for arrays"

More than the allowed three dimensions are given in the definition of an array. If applicable use an ARRAY OF ARRAY.

3749

"lower bound '<name>' not defined"

There is a not defined constant used to define the lower limit for a subrange or array type.

3750

"upper bound '<name>' not defined"

There is a not defined constant used to define the upper limit for a subrange or array type.

3751

"Invalid string length '<number of characters>'"

The here defined string length exceeds the maximum value which is defined for the currently set target system.

3752

"More than 9 dimensions are not allowed for nested arrays"

An array can be 1- 2- or 3-dimensional. The number of dimensions reached by nesting of arrays (e.g "arr: ARRAY [0..2,0..2,0..2] OF ARRAY [0..2,0..2,0..2] OF ARRAY [0..2,0..2,0..2, 0..2] OF DINT" maximum may be 9 and is exceeded in the current error case. Reduce appropriately to max. 9 dimensions.

3760

"Error in initial value"

Use an initial value which corresponds to the type definition. To change the declaration you can use the declaration dialog for variables (Shift/F2 or 'Edit'Autodeclare').

3761

""VAR_IN_OUT' variables must not have an initial value."

Remove the initialization at the declaration of the VAR_IN_OUT variable.

3780

""VAR', 'VAR_INPUT', 'VAR_OUTPUT' or 'VAR_IN_OUT' expected"

The first line following the name of a POU must contain one of these keywords.

3781

""END_VAR' or identifier expected"

Enter a valid identifier of a END_VAR at the beginning of the given line in the declaration window.

3782

"Unexpected end"

In the declaration editor: Add keyword END_VAR at the end of the declaration part.

In the texteditor of the programming part: Add an instruction which terminates the last instruction sequence (e.g. END_IF).

3783

"END_STRUCT' or identifier expected"

Ensure that the type declaration is terminated correctly.

- 3784**
"The current target doesn't support attribute <attribute name>"
 The target system does not support this type of variables (e.g. RETAIN, PERSISTENT)
- 3800**
"The global variables need too much memory. Increase the available memory in the project options."
 Increase the number of segments given in the settings in dialog ,Project' ,Options' ,Build'.
- 3801**
"The variable '<Name>' is too big. (<size> byte)"
 The variable uses a type which is bigger than 1 data segment. The segment size is a target specific parameter and can be modified in the target settings/memory layout. If you do not find this in the current target settings, please contact your PLC manufacturer.
- 3802**
"Out of retain memory. Variable '<name>', <number> bytes."
 The memory space available for Retain variables is exhausted. The size of the memory area can be set target-specific in the target settings /memory layout. If you do not find the settings field in the dialog, please contact your PLC manufacturer. (Please regard: If retain variables are used in an function block instance, the complete instance POU will be stored in the retain memory area !)
- 3803**
"Out of global data memory. Variable '<name>', ,<number>' bytes."
 The memory space available for global variables is exhausted. The size of the memory area can be set target-specific in the target settings /memory layout. If you do not find the settings field in the dialog, please contact your PLC manufacturer.
- 3804**
"The current size of the persistent data description is '<number>' bytes and exceeds the maximum of '<number>' bytes."
 Reduce the persistent data to that size which is allowed by the target system.
- 3820**
"'VAR_OUTPUT' and 'VAR_IN_OUT' not allowed in functions"
 In a function no output or in_output variables may be defined.
- 3821**
"At least one input required for functions"
 Add at least on input parameter for the function.
- 3840**
"Unknown global variable '<name>!'"
 In the POU a VAR_EXTERNAL variable is used, for which no global variable declared.
- 3841**
"Declaration of '<name>' do not match global declaration!"
 The type given in the declaration of the VAR_EXTERNAL variable is not the same as that in the global declaration.

- 3850**
"Declaration of an unpacked struct '<name>' inside a packed struct '<name>' is not allowed!"
This structure definition leads to misalignment in the memory. Modify the definition appropriately.
- 3900**
"Multiple underlines in identifier"
Remove multiple underlines in the identifier name.
- 3901**
"At most 4 numerical fields allowed in addresses"
There is a direct assignment to an address which has more than four levels. (e.g. %QB0.1.1.0.1).
- 3902**
"Keywords must be uppercase"
Use capital letters for the keyword or activate option ,Autoformat' in ,Project' ,Options'.
- 3903**
"Invalid duration constant"
The notation of the constant does not comply with the IEC61131-3 format.
- 3904**
"Overflow in duration constant"
The value used for the time constant cannot be represented in the internal format. The maximum value which is presentable is t#49d17h2m47s295ms.
- 3905**
"Invalid date constant"
The notation of the constant does not comply with the IEC61131-3 format.
- 3906**
"Invalid time of day constant"
The notation of the constant does not comply with the IEC61131-3 format.
- 3907**
"Invalid date and time constant"
The notation of the constant does not comply with the IEC61131-3 format.
- 3908**
"Invalid string constant"
The string constant contains an invalid character.
- 4000**
"Identifier expected"
Enter a valid identifier at this position.
- 4001**
"Variable '<Name>' not declared"
Declare variable local or global.

- 4010**
"Type mismatch: Cannot convert '<Name>' to '<Name>'."
 Check what data type the operator expects (Browse Online Help for name of operator) and change the type of the variable which has caused the error, or select another variable.
- 4011**
"Type mismatch in parameter '<Name>' of '<Name>': Cannot convert '<Name>' to '<Name>'."
 The data type of the actual parameter cannot be automatically converted to that of the formal parameter. Use a type conversion or use another variable type.
- 4012**
"Type mismatch in parameter '<Name>' of '<Name>': Cannot convert '<Name>' to '<Name>'."
 A value with the invalid type <Typ2> is assigned to the input variable '<Name>'. Replace the variable or constant to one of type <Typ1> or use a type conversion respectively a constant with type-prefix.
- 4013**
"Type mismatch in output '<Name>' of '<Name>': Cannot convert '<Name>' to '<Name>'."
 A value with the invalid type <Typ2> is assigned to the output variable '<Name>'. Replace the variable or constant to one of type <Typ1> or use a type conversion respectively a constant with type-prefix.
- 4014**
"Typed literal: Cannot convert '<name>' to '<name>'"
 The type of the constant is not compatible with the type of the prefix.
 Example: SINT#255
- 4015**
"Data type '<name>' illegal for direct bit access"
 Direct bit addressing is only allowed for Integer- Bitstring datatypes and not at all for direct variables. You are using a variable var1 of type Typ REAL/LREAL or a constant in bit access <var1>.<bit>, or you are trying a bit access on a direct variable.
- 4016**
"Bit index '<number>' out of range for variable of type '<name>'"
 You are trying to access a bit which is not defined for the data type of the variable.
- 4017**
""MOD' is not defined for 'REAL'"
 The operator MOD can only be used for integer and bitstring data types.
- 4020**
"Variable with write access or direct address required for 'ST', 'STN', 'S', 'R'"
 Replace the first operand by a variable with write access.
- 4021**
"No write access to variable '<name>' allowed"
 Replace the variable by a variable with write access.
- 4022**
"Operand expected"
 Add an operand behind the command.

4023

"Number expected after '+' or '-"

Enter a digit.

4024

"Expecting <Operator 0> or <Operator 1> or ... before '<Name>'"

Enter a valid operand at the named position.

4025

"Expecting ':=' or '=>' before '<Name>'"

Enter one of the both operators at the named position.

4026

"'BITADR' expects a bit address or a variable on a bit address"

Use a valid bit address (e.g. %IX0.1).

4027

"Integer number or symbolic constant expected"

Enter a integer number or the identifier of a valid constant.

4028

"'INI' operator needs function block instance or data unit type instance"

Check the data type of the variable, for which the INI operator is used.

4029

"Nested calls of the same function are not possible."

At not reentrant target systems and in simulation mode a function call may not contain a call of itself as a parameter.

Example: fun1(a,fun1(b,c,d),e);

Use an intermediate table.

4030

"Expressions and constants are not allowed as operands of 'ADR'"

Replace the constant or the expression by a variable or a direct address.

4031

"'ADR' is not allowed on bits! Use 'BITADR' instead."

Use BITADR. Please note: The BITADR function does not return a physical memory address.

4032

"'<number>' operands are too few for '<name>'. At least '<number>' are needed"

Check how many operands the named operator requires and add the missing operands.

4033

"'<number>' operands are too many for '<name>'. At least '<number>' are needed"

Check how many operands the named operator requires and remove the surplus operands.

- 4034**
"Division by 0"
 You are using a division by 0 in a constant expression. If you want to provoke a runtime error, use – if applicable - a variable with the value 0.
- 4035**
"ADR must not be applied on 'VAR CONSTANT' if 'replaced constants' is activated"
 An address access on constants for which the direct values are used, is not possible. If applicable, deactivate the option ‚Replace Constants‘ in ‚Project‘ ‚Options‘ ‚Build‘.
- 4040**
"Label '<name>' is not defined"
 Define a label with the name <LabelName> or change the name <LabelName> to that of a defined label.
- 4041**
"Duplicate definition of label '<name>'"
 The label '<Name>' is multiple defined in the POU. Rename the label or remove one of the definitions.
- 4042**
"No more than <number> labels in sequence are allowed"
 The number of jump labels is limited to '<Anzahl>'. Insert a dummy instruction.
- 4043**
"Format of label invalid. A label must be a name optionally followed by a colon."
 "The label name is not valid or the colon is missing in the definition.
- 4050**
"POU '<name>' is not defined"
 Define a POU with the name '<Name>' using the command **'Project' 'Add Object'** or change '<Name>' to the name of a defined POU.
- 4051**
"'<name>' is no function"
 Use instead of <Name> a function name which is defined in the project or in the libraries.
- 4052**
"'<name>' must be a declared instance of FB '<name>'"
 Use an instance of data type '<Name>' which is defined in the project or change the type of <Instance name> to '<Name>'.
- 4053**
"'<name>' is no valid box or operator"
 Replace '<Name>' by the name of a POU or an operator defined in the project.
- 4054**
"POU name expected as parameter of 'INDEXOF'"
 The given parameter is not a valid POU name.

4060

""VAR_IN_OUT' parameter '<name>' of '<name>' needs variable with write access as input"

To VAR_IN_OUT parameters variables with write access have to be handed over, because a VAR_IN_OUT can be modified within the POU.

4061

""VAR_IN_OUT' parameter '<name>' of '<name>' must be used."

A VAR_IN_OUT parameter must get handed over a variable with write access, because a VAR_IN_OUT can be modified within the POU.

4062

"No external access to 'VAR_IN_OUT' parameter '<name>' of '<name>'."

VAR_IN_OUT Parameter only may be written or read within the POU, because they are handed over by reference.

4063

""VAR_IN_OUT' parameter '<name>' of '<name>' must not be used with bit addresses."

A bit address is not a valid physical address. Hand over a variable or a direct non-bit address.

4064

""VAR_IN_OUT' must not be overwritten in local action call!"

Delete the parameters set for the VAR_IN_OUT variable in the local action call.

4070

"The POU contains a too complex expression"

Decrease nesting depth by dividing up the expression into several expressions. Use intermediate variables for this purpose.

4071

"Network too complex"

Divide up the network into several networks.

4072

"Inconsistent use of an action identifier in FB type ('<name>') and instance ('<name>')."

You have defined two actions of a function block fb: e.g. a1 and a2, but in the call of one of the actions in the FBD you are using a type (string within the box, e.g. fb.a1 different to that used in the instance-name (e.g. inst.a2, above box). Correct the name correspondingly into the name of the desired action.

4100

""^' needs a pointer type"

You are trying to dereference a variable which is not declared as a pointer.

4110

""[<index>]' needs array variable"

[<index>] is used for a variable which is not declared as an array with ARRAY OF.

4111

"Index expression of an array must be of type 'INT'"

Use an expression of the correct type or a type conversion.

- 4112**
"Too many indexes for array"
Check the number of indices (1, 2, or 3), for which the array is declared and remove the surplus.
- 4113**
"Too few indexes for array"
Check the number of indices (1, 2, or 3), for which the array is declared and add the missing ones.
- 4114**
"One of the constant indices is not within the array range"
Make sure that the used indices are within the bounds of the array.
- 4120**
"".' needs structure variable""
The identifier on the left hand of the dot must be a variable of type STRUCT or FUNCTION_BLOCK or the name of a FUNCTION or a PROGRAM.
- 4121**
"" '<Name>' is not a component of <object name>""
The component '<Name>' is not included in the definition of the object <object name>.
- 4122**
""<name>' is not an input variable of the called function block""
Check the input variables of the called function block and change '<name>' to one of these.
- 4200**
""LD' expected""
Insert at least one LD instruction after the jump label in the IL editor.
- 4201**
""IL Operator expected""
Each IL instruction must start with an operator or a jump label.
- 4202**
""Unexpected end of text in brackets""
Insert a closing bracket after the text.
- 4203**
""<Name> in brackets not allowed""
The operator <name> is not valid in a IL bracket expression.
(not valid are: 'JMP', 'RET', 'CAL', 'LDN', 'LD', 'TIME')
- 4204**
""Closing bracket with no corresponding opening bracket""
Insert an opening bracket or remove the closing one.
- 4205**
""No comma allowed after)""
Remove comma after closing bracket.

4206

"Label in brackets not allowed"

Shift jump label so that it is outside of the brackets.

4207

"'N' modifier requires operand of type 'BOOL','BYTE','WORD' or 'DWORD'"

The N modifier requires a data type, for which a boolean negation can be executed.

4208

"Conditional Operator requires type 'BOOL'"

Make sure that the expression gives out a boolean result or use a type conversion.

4209

"Function name not allowed here"

Replace the function call by a variable or a constant.

4210

"'CAL', 'CALC' and 'CALN' require a function block instance as operand"

Declare an instance of the function block which you want to call.

4211

"Comments are only allowed at the end of line in IL"

Shift the comment to the end of the line or to an extra line.

4212

"Accumulator is invalid before conditional statement"

The accu is not defined. This happens if an instruction is preceding which does not submit a result (e.g. 'CAL').

4213

"'S' and 'R' require 'BOOL' operand"

Use a boolean variable at this place.

4250

"Another 'ST' statement or end of POU expected"

The line does not start with a valid ST instruction.

4251

"Too many parameters in function '<name>'"

There are more parameters given than are declared in the definition of the function.

4252

"Too few parameters in function '<name>'"

There are less parameters given than are declared in the definition of the function.

4253

"'IF' or 'ELSIF' require 'BOOL' expression as condition"

Make sure that the condition for IF or ELSIF is a boolean expression.

- 4254**
""WHILE' requires 'BOOL' expression as condition"
Make sure that the condition following the 'WHILE' is a boolean expression.
- 4255**
""UNTIL' requires 'BOOL' expression as condition"
Make sure that the condition following the 'UNTIL' is a boolean expression.
- 4256**
""NOT' requires 'BOOL' operand"
Make sure that the condition following the 'NOT' is a boolean expression.
- 4257**
"Variable of 'FOR' statement must be of type 'INT'"
Make sure that the counter variable is of an integer or bitstring data type (e.g. DINT, DWORD).
- 4258**
"Expression in 'FOR' statement is no variable with write access"
Replace the counter variable by a variable with write access.
- 4259**
"Start value in 'FOR' statement is no variable with write access"
The start value in the ,FOR' instruction must be compatible to the type of the counter variable.
- 4260**
"End value of 'FOR' statement must be of type 'INT'"
The end value in the ,FOR' instruction must be compatible to the type of the counter variable.
- 4261**
"Increment value of 'FOR' statement must be of type 'INT'"
The incremental value in the ,FOR' instruction must be compatible to the type of the counter variable.
- 4262**
""EXIT' outside a loop"
Use 'EXIT' only within 'FOR', 'WHILE' or 'UNTIL' instructions.
- 4263**
"Expecting Number, 'ELSE' or 'END_CASE'"
Within a 'CASE' expression you only can use a number or a 'ELSE' instruction or the ending instruction 'END_CASE'.
- 4264**
""CASE' requires selector of an integer type"
Make sure that the selector is of an integer or bitstring data type (e.g. DINT, DWORD).
- 4265**
"Number expected after ','"
In the enumeration of the CASE selectors there must be inserted a further selector after a comma.

4266

"At least one statement is required"

Insert an instruction, at least a semicolon.

4267

"Function block call requires function block instance"

The identifier in the function block call is no instance. Declare an instance of the desired functionblock or use the name of an already defined instance.

4268

"Expression expected"

Insert an expression.

4269

""END_CASE' expected after 'ELSE'-branch"

Terminate the 'CASE' instruction after the 'ELSE' part with an 'END_CASE'.

4270

""CASE' constant '<name>' already used"

A 'CASE' selector may only be used once within a 'CASE' instruction.

4271

"The lower border of the range is greater than the upper border."

Modify the area bounds for the selectors so that the lower border is not higher than the upper border.

4272

"Expecting parameter '<name>' at place <position> in call of '<name>'!"

You can edit a function call in that way, that also the parameter names are contained, not only the parameter values. But nevertheless the position (sequence) of the parameters must be the same as in the function definition.

4273

Parts of the 'CASE'-Range '<range>' already used in Range '<range>'

Make sure that the areas for the selectors which are used in the CASE instruction, don't overlap.

4274

"Multiple 'ELSE' branch in 'CASE' statement"

A CASE instruction may not contain more than one ,ELSE' instruction.

4300

"Jump requires 'BOOL' as input type"

Make sure that the input for the jump respectively the RETURN instruction is a boolean expression.

4301

"POU '<name>' need exactly <number> inputs"

The number of inputs does not correspond to the number of VAR_INPUT and VAR_IN_OUT variables which is given in the POU definition.

- 4302**
"POU '<name>' need exactly %d outputs".
The number of outputs does not correspond to the number of VAR_OUTPUT variables which is given in the POU definition.
- 4303**
""<name>' is no operator"
Replace '<Name>' by a valid operator.
- 4320**
"Non-boolean expression '<name>' used with contact"
The switch signal for a contact must be a boolean expression.
- 4321**
"Non-boolean expression '<name>' used with coil"
The output variable of a coil must be of type BOOL.
- 4330**
"Expression expected at input 'EN' of the box '<name>' "
Assign an input or an expression to the input EN of POU '<Name>'.
- 4331**
"Expression expected at input '<number>' of the box '<Name>' "
The input <number> of the operator POU is not assigned.
- 4332**
"Expression expected at input '<name>' of the box '<Name>'"
The input of the POU is of type VAR_IN_OUT and is not assigned.
- 4333**
"Identifier in jump expected"
The given jump mark is not a valid identifier.
- 4334**
"Expression expected at the input of jump"
Assign a boolean expression to the input of the jump. If this is TRUE, the jump will be executed.
- 4335**
"Expression expected at the input of the return"
Assign a boolean expression to the input of the RETURN instruction. If this is TRUE, the jump will be executed.
- 4336**
"Expression expected at the input of the output"
Assign a suitable expression to the output box.
- 4337**
"Identifier for input expected"
Insert a valid expression or identifier in the input box.

4338

"Box '<name>' has no inputs"

To none of the inputs of the operator POU '<Name>' a valid expression is assigned.

4339

"Typemismatch at output: Cannot convert '<name>' to '<name>'."

The type of the expression in the output box is not compatible to that of the expression which should be assigned to it.

4340

"Jump requires 'BOOL' as input type"

Make sure that the input for the jump is a boolean expression.

4341

"Return needs a boolean input"

Make sure that the input for the RETURN instruction is a boolean expression.

4342

"Expression expected at input 'EN' of the box '<name>'"

Assign a valid boolean expression to the EN input of the box.

4343

"Values of Constants: '<name>'"

Input '<Name>' of box '<Name>' is declared as VAR_INPUT CONSTANT. But to this POU box an expression has been assigned in the dialog 'Edit Parameters' which is not type compatible.

4344

"'S' and 'R' require 'BOOL' operand"

Insert a valid boolean expression after the Set resp. Reset instruction.

4345

"Invalid type for parameter '<name>' of '<name>': Cannot convert '<type>' to '<type>'."

An expression is assigned to input '<Name>' of POU box '<Name>' which is not type compatible.

4346

"Not allowed to use a constant as an output"

You can only assign an output to a variable or a direct address with write access.

4347

"'VAR_IN_OUT' parameter needs variable with write access as input"

To VAR_IN_OUT parameters only variables with write access can be handed over, because these can be modified within the POU.

4348

"Invalid program name '<name>'. A variable with the same name exists already."

You have inserted a program box in the CFC editor, which has the same name as a (global) variable already existing in your project. You must rename accordingly.

- 4349**
“Input or output in POU <name> has been deleted: Check all connections to the box. This error message disappears only after the CFC was edited”
Correct the POU appropriately.
- 4350**
"An SFC-Action can not be accessed from outside!"
SFC actions only can be called within the SFC POU in which they are defined. But this error also will be dumped, if you call an action from within a SFC POU, which is allowed, but are not using IEC steps while the iecsfc.lib is still included in your project. In this case please remove the library in the library manager and rebuild the project.
- 4351**
"Step name is no identifier: '<name>'"
Rename the step or choose a valid identifier as step name.
- 4352**
"Extra characters following valid step name: '<Name>'"
Remove the not valid characters in the step name.
- 4353**
"Step name duplicated: '<Name>'"
Rename one of the steps.
- 4354**
"Jump to undefined Step: '<Name>'"
Choose an existent step name as aim of the jump resp. insert a step with name ,<name>'.
- 4355**
"A transition must not have any side effects (Assignments, FB-Calls etc.)"
A transition must be a boolean expression.
- 4356**
"Jump without valid Step Name: '<Name>' "
Use a valid identifier as aim (mark) of the jump.
- 4357**
"IEC-Library not found"
Check whether the library iecsfc.lib is inserted in the library manager and whether the library paths defined in 'Project' 'Options' 'Paths' are correct.
- 4358**
"Action not declared: '<name>'"
Make sure that in the object organizer the action of the IEC step is inserted below the SFC POU and that in the editor the action name is inserted in the box on the right hand of the qualifier.
- 4359**
"Invalid Qualifier: '<name>'"
In the box on the left hand of the action name enter a qualifier for the IEC action.

4360

"Time Constant expected after qualifier '<name>'"

Enter next to the box on the left hand of the action name a time constant behind the qualifier.

4361

"'<name>' is not the name of an action"

Enter next to the box on the right hand of the qualifier the name of an action or the name of a variable which is defined in the project.

4362

"Nonboolean expression used in action: '<name>'"

Insert a boolean variable or a valid action name.

4363

"IEC-Step name already used for variable: '<Name>'"

Please rename the step or the variable.

4364

"A transition must be a boolean expression"

The result of the transition expression must be of type BOOL.

4365

"Time Constant expected after qualifier '<name>'"

Open dialog ,step attributes' for the step '<Name>' and enter a valid time variable or time constant.

4366

"The label of the parallel branch is no valid identifier: '<Name>'"

Enter a valid identifier next to the triangle which marks the jump label.

4367

"The label '<name>' is already used"

There is already a jump label or a step with this name. Please rename correspondingly.

4368

"Action '<name>' is used in multiple step chains, where one is containing the other!"

The action '<Name>' is used in the POU as well as in one or several actions of the POU.

4369

"Exactly one network required for a transition"

There are used several FBD resp. LD networks for a transition. Please reduce to 1 network.

4370

"Additional lines found after correct IL-transition"

Remove the not needed lines at the end of the transition.

4371

"Invalid characters following valid expression: '<name>'"

Remove the not needed characters at the end of the transition.

- 4372**
"Step '<name>': Time limit needs type 'TIME'"
Define the time limits of the step in the step attributes by using a variable of type TIME or by a time definition in correct format (e.g. "t#200ms").
- 4373**
"IEC-actions are only allowed with SFC-POUs"
There is an action assigned to a non-SFC-POU (see in the Object Organizer), which is programmed in SFC and which contains IEC actions. Replace this action by one which contains no IEC actions.
- 4374**
"Step expected instead of transition '<name>'"
The SFC POU is corrupt, possibly due to any export-import actions.
- 4375**
"Transition expected instead of step '<name>'"
The SFC POU is corrupt, possibly due to any export-import actions.
- 4376**
"Step expected after transition '<name>'"
The SFC POU is corrupt, possibly due to any export-import actions.
- 4377**
"Transition expected after step '<name>'"
The SFC POU is corrupt, possibly due to any export-import actions.
- 4400**
Import / conversion of POU '<name>' contains errors resp. is not complete."
The POU cannot be converted to IEC 61131-3 completely.
- 4401**
"S5 time constant <number> seconds is too big (max. 9990s)."
There is no valid BCD coded time in the accu.
- 4402**
"Direct access only allowed on I/Os."
Make sure that you only access variables which are defined as input or output.
- 4403**
"STEP5/7 instruction invalid or not convertible to IEC 61131-3."
Some STEP5/7 commands are not convertible to IEC 61131-3, e.g. CPU commands like MAS.
- 4404**
"STEP5/7 operand invalid or not convertible to IEC 61131-3."
Some STEP5/7 operands are not convertible to IEC 61131-3 respectively an operand is missing.
- 4405**
"Reset of a STEP5/7 timer cannot be converted into IEC 61131-3."
The corresponding IEC timer have no reset input.

4406

"STEP5/7 Counter constant out of range (max. 999)."

There is no valid BCD coded counter constant in the accu.

4407

"STEP5 instruction not convertible to IEC 61131-3."

Some STEP5/7 instructions cannot be converted to IEC 61131-3, e.g. DUF.

4408

"Bit access of timer or counter words not convertible into IEC 61131-3."

Special timer/counter commands are not convertible into IEC 61131-3.

4409

"Contents of ACCU1 or ACCU2 undefined, not convertible into IEC 61131-3."

A command, which connects the both accus, cannot be converted, because the accu values are not defined.

4410

"Called POU not in project."

Import the called POU.

4411

"Error in global variable list."

Please check the SEQ file.

4412

"Internal error no.11"

Please contact the PLC manufacturer.

4413

"Error in format of line in data block"

In the code which should be imported there is an erroneous date.

4414

"FB/FX name missing."

In the original S5D file the symbolic name of an (extended) POU is missing.

4415

"Instruction after block end not allowed."

A protected POU cannot get imported.

4416

"Invalid command"

The S5/S7 command cannot be disassembled.

4417

"Comment not closed"

Close the comment with "*)".

- 4418**
"FB/FX-Name too long (max. 8 characters)"
The symbolic name of an (extended) POU is too long.
- 4419**
"Expected format of line "(* Name: <FB/FX-Name> *)" "
Correct the line correspondingly.
- 4420**
"Name of FB/FX parameter missing"
Check the POUs.
- 4421**
"Type of FB/FX parameter invalid"
Check the POUs.
- 4422**
"Type of FB/FX parameter missing"
Check the POUs.
- 4423**
"Invalid FB/FX call parameter"
Check the interface of the POU.
- 4424**
"Warning: FB/FX for call either missing or parameters invalid or has '0' parameters"
The called POU is not imported yet or is not correct or has no parameters (in the last case you can ignore the error message).
- 4425**
"Definition of label missing"
The aim (label) of the jump is not defined.
- 4426**
"POU does not have a valid STEP 5 block name, e.g. PB10"
Modify the POU name.
- 4427**
"Timer type not declared"
Add a declaration of the timer in the global variables list.
- 4428**
"Maximum number of open STEP5 brackets exceeded"
You may not use more than seven open brackets.
- 4429**
"Error in name of formal parameter"
The parameter name may not exceed four characters.

4430

"Type of formal parameter not IEC-convertible"

In IEC 61131-3 Timer, counter and POU's cannot be converted as formal parameters.

4431

"Too many 'VAR_OUTPUT' parameters for a call in STEP5 STL"

A POU may not contain more than 16 formal parameters as outputs.

4432

"Labels within an expression are not allowed"

In IEC 61131-3 jump labels may not be inserted at any desired position.

4434

"Too many labels"

A POU may not contain more than 100 labels.

4435

"After jump / call, a new expression must start"

After jump or call a Load command LD must follow.

4436

"Bit result undefined, not convertible to IEC 61131-3."

The command which is used by VKE cannot get converted, because the value of the VKE is not known.

4437

"Type of instruction and operand are not compatible"

A bit command is used for a word operand or the other way round.

4438

"No data block opened (insert instruction C DB before)"

Insert a "A DB".

4500

"Unrecognized variable or address"

The watch variable is not declared within the project. By pressing <F2> you get the input assistant which lists the declared variables.

4501

"Extra characters following valid watch expression"

Remove the surplus signs.

4520

"Error in Pragma: Flag expected before '<Name>'"

The pragma is not correct. Check whether '<Name>' is a valid flag.

4521

"Error in Pragma: Unexpected element '<Name>'"

Check whether pragma is composed correctly.

- 4522**
"flag off" pragma expected!"
Pragma has not been terminated, insert a 'flag off' instruction.
- 4523**
"Pragma {<Pragmaname>} not allowed in interface of type '<Name>'"
Das Pragma cannot be used at this location. Please see the CoDeSys Online Help resp. Users Guide, keyword 'pragma' for the correct use of pragmas.
- 4550**
"Index out of defined range : Variable OD "number", Line <line number>."
Ensure that the index is within the area which is defined in the target settings/networkfunctionality.
- 4551**
"Subindex out of defined range : Variable OD "number", Line <line number>."
Ensure that the subindex is within the area which is defined in the target settings/networkfunctionality.
- 4552**
"Index out of defined range : Parameter OD "number", Line <line number>."
Ensure that the index is within the area which is defined in the target settings/networkfunctionality.
- 4553**
"Subindex out of defined range : Parameter OD "number", Line <line number>."
Ensure that the subindex is within the area which is defined in the target settings/networkfunctionality.
- 4554**
"Variablename invalid: Variable OD <number>, Line <line number>."
Enter a valid project variable in the filed ,variable'. Use the syntax <POU name>.<variable name> resp. for global variables .<variable name>
- 4555**
"Empty table-entry, input not optional: Parameter OD <number>, Line <line number>"
You must make an entry in this field.
- 4556**
"Empty table-entry, input not optional: Variable OD <number>, Line <number>"
You must make an entry in this field.
- 4557**
"The required parameter memory is too large"
The maximum size of data which can be loaded via parameter lists of type Parameters to the controller has been exceeded. This size is defined by the target system. Information on the data size is displayed in the message window at compilation. Reduce the parameter lists size.
- 4558**
"The required variable memory is too large"
The maximum size of data which can be loaded via parameter lists of type Variables to the controller has been exceeded. This size is defined by the target system. Information on the data size is displayed in the message window at compilation. Reduce the parameter lists size.

4560

"Invalid value: Dictionary '<Name>', column '<Name>', line '<line number>'"

Check this entry. It depends on the currently used column (attribute) definition which entries are valid for this field. This definition is given by the target-specific XML description file of the Parameter Manager resp. by the standard settings which will be used if there is no description file.

4561

"Column not defined: '<Name>'"

Entries in a column of the parameter list refer to another column, which is not defined however. The column definitions are given by the description file (XML) of the Parameter Manager for the current target. If a description file is not available, standard settings are used.

4562

"Index/subindex used already: Dictionary '<Name>', line '<Line Number>'"

The Index/Subindex-combination must be unique throughout all parameter lists, because it can be used for the parameter access. Correct the indices correspondingly.

4563

"Identifier '<Name>' used already: Dictionary '<Name>', line '<Line Number>'"

The name must be unique throughout all parameter lists, because it can be used for parameter access.

4564

"Index '<Name>' is out of range: Dictionary '<Name>', line '<Line Number>' "

Enter an index which is within the range defined in the target settings, category network functionality in field 'Index range...' for the respective list types (Variables, Parameters, Mappings).

4565

"Subindex '<Name>' is out of range: Dictionary '<Name>', line '<Line Number>' "

Enter an subindex which is within the range defined in the target settings, category network functionality in field 'SubIndex range'.

4566

"An error occurred during import of the parameter manager"

You have imported an export file which contains erroneous information on the Parameter Manager. Check the *.exp-file.

4600

"Networkvariables: '<name>' expression is not from type bool!"

Make sure that the variable defined in the properties dialog of the network variables list at option 'Transmit on event', is of type BOOL.

4601

"Network variables '<name>': No cyclic or freewheeling task for network variable exchange found"

There is no cyclic or free-wheeling task resp. PLC_PRG in the project where the network variables of type CAN or UDP of the given list are used (only declaration is not sufficient!). You must take care that the variables are used in an appropriate task or in PLC_PRG. If you want to use them in several tasks, regard, that at data exchange the task with the highest priority will be regarded.

- 4602**
""<name of network variables list>: The object uses UDP port '<port number>' instead of '<port number>'"
 In the Settings of the named network variables list a port number is used which is not the same as that which is used in the first network variables list found in the global variables folder. Take care that all network variables lists are using the same port!
- 4604**
"Network variables '<name>': Base identifier has been used more than once."
 The same COB-ID is used in the configuration settings ('Object' 'Properties') of multiple Network Variables lists. Assign unique IDs.
- 4605**
"Network variables '<name>': Duplicate CAN COB id."
 In the configuration of a Network Variables list ('Object' 'Properties') a COB-ID is used, which is also specified in the CAN PLC Configuration. Assign unique IDs.
- 4620**
 Unused variables have been found in the project. See See Chapter 4.3 for the description for command 'Project' 'Check' Unused Variables.
- 4621**
 There are overlaps at the assignment of variables to memory areas via the „AT“-declaration. See Chapter 4.3 for the description of command 'Project' 'Check' 'Overlapping memory areas'.
- 4622**
 IEC addresses assigned to the same memory area are referenced in more than one task. See Chapter 4.3 for the description of command 'Project' 'Check' 'Concurrent Access'.
- 4623**
 The project gains write access to the same memory area at more than one place. See Chapter 4.3 for the description of command 'Project' 'Check' 'Multiple writes to output'.
- 4650**
"AxisGroup '<Name>': Task '<Name>' does not exist."
 In the PLC Configuration in the definition of the axis group (dialog 'Module parameters', column 'Value') there is a name defined for the task which is controlling the data transfer of this axis group, which is not known in the Task Configuration. Correct Task Configuration resp. PLC Configuration correspondingly.
- 4651**
"AxisGroup '<Name>': Cycletime (dwCycle) not set."
 In dialog 'Module parameters' of the axis group enter a value for the cycle time (dwCycle).
- 4656**
"Drive '<name>': Scaling factor must not be 0."
 Open the PLC Configuration and check the Scale entries in the configuration dialog of the drive. All fields must have entries not equal "0".
- 4670**
"CNC program '<Name>': Global variable '<Name>' not found."

In the CNC program a global variable is used (e.g. \$glob_var\$), which is not defined in the project. Add the appropriate declaration resp. correct the assignment to the variable in the CNC program.

4671

"CNC program '<Name>': Variable '<Name>' has an incompatible type."

There is a variable assigned in a instruction of the CNC program , which is declared of a data type which is not valid in this place. Use another variable resp. correct the type specification.

4685

"CAM '<Name>': CAM table type unknown."

Check the data type which is specified in the CAM Editor dialog "Compile options.." for the equidistant resp. element optimized point table.

4686

"CAM '<Name>': CAM point exceeds datatype range."

In this CAM points are used, which are out of the data range specified for the point table. For the current range definition see dialog 'Compile options..' in the CAM-Editor.

4700

"'<Number>' ('<Name>'): Watch expression '<Name>' is not a numeric variable."

In the configuration of the visualization a variable is used which is not a number, as required in this place (e.g. at the configuration of XOffset or Angle values etc.).

4701

"'<Name>' ('<Number>'): Watch expression '<name>' is not of type BOOL."

In the configuration of the visualization a variable is used which is not of type BOOL, as required in this place.

4702

"'<Name>' ('<Number>'): Watch expression '<name>' is not of type STRING."

The visualization contains a variable which is not of type STRING although this is required in this place (e.g. at the tooltip configuration).

4703

"'<Name>' ('<Number>') : Invalid watch expression '<Name>'"

The visualization contains an invalid variable.

4704

"'<Name>' ('<Number>'): Invalid initial value in watchlist '<Name>'."

In this watchlist, used in a visualization (INTERN command in category Input), there is a erroneous initial value. Check the used list.

4705

"'<Name>' ('<Number>'): No valid alarm group assigned to alarm table."

Enter a valid alarm group In the configuration dialog of the alarm table (category Alarm table).

4706

"'<Name>' ('<Number>'): Use of alarm tables requires the target setting 'Alarmhandling in controller' to be activated.."

Open the target settings in tab 'Resources' and in dialog 'Visualization' activate option 'Alarmhandling in the PLC'. Otherwise the alarm table element will not work in the Target-Visualization which is currently activated also in the target settings.

4707

""<Name>' (<Number>'): Alarm tables are not supported by the current target. Please remove these elements from the target visualization."

The target system does not support the processing of alarms (target setting 'Alarmhandling in the PLC' cannot be activated). Thus for running a Target-Visualization (currently activated in the target settings in tab 'Visualization') the alarm table elements must be removed from the visualization.

4708

""<Name>' (<Number>'): Use of trends requires the target setting 'Trend recording in the PLC' to be activated."

Open the target settings in tab 'Resources' and in dialog 'Visualization' activate option 'Store trend data in the PLC'. Otherwise the trend element will not work in the Target-Visualization which is currently activated also in the target settings.

4709

""<Name>' (<Number>'): Trends are not supported by the current target. Please remove these elements from the target visualization."

The target system does not support the processing of trend data (target setting 'Store trend data in the PLC' cannot be activated). Thus for running a Target-Visualization (currently activated in the target settings in tab 'Visualization') the trend elements must be removed from the visualization.

4712

""<Name>' (<number>'): The selected target does not support the output of passwords""

The text output in the visualization element is configured to work via dynamic texts or unicode texts. Additionally the option "Hidden" is activated in the "Input" category. This is not supported by the Target-Visualization.

4900

"Invalid type for conversion"

You are using a type conversion which is not supported by the currently chosen codegenerator.

4901

"Internal error: Overflow in array access!"

The array bounds are too large for a 32-bit-variable. Reduce the array index range.

11 Index

8

8.3 file format 10-97
8051 10-91

A

Accept access rights 4-42
Accept change 4-41
Accept changed item 4-41
Accept properties 4-41
Access conflict 4-45
Access protection 4-24, 4-46
Access right 4-46
Access right of DeviceNet-Slave parameter 6-49
Access rights 4-58
acknowledgement 6-10
Acknowledgement of alarms 6-9
Action
 Associate in SFC 5-45
Action 2-7, 2-16, 4-59
Action in SFC
 Add 5-41
 Zoom 5-42
Action on timeout error 6-48
Actions hide programs 4-12
Activate heartbeat consumer 6-43
Activate Heartbeat Consumer 6-39
Activate heartbeat generation 6-43
Activate Heartbeat Producer 6-39
Activate system variable CurrentVisu 10-97
Activation 6-33
Active step 2-17
ADD 10-1
Add configuration file 6-24
Add Label to parallel branch 5-41
Add Object 4-54, 4-55
ADD Operator in AWL 2-9
Add Shared Objects 4-52
Additional CoDeSys Features 1-2
Additional Online Functions 1-1
Address
 DeviceNet-Master 6-45
 DeviceNet-Slave 6-46
Address check for PLC configuration 10-94
Addresses 10-29
ADR 10-13
Address of an instance 10-13
ADRINST 10-13
Alarm acknowledgement 6-10
Alarm class 6-13
Alarm classes 6-10
Alarm configuration 6-10, 6-13
Alarm configuration settings 6-16
Alarm configuration\Online settings 6-16
Alarm deactivation 6-13
Alarm evaluation deactivation 6-16
Alarm Event 6-9
Alarm group 6-13
Alarm message 6-13
Alarm priority 6-9, 6-13
Alarm saving 6-15
Alarm state 6-9

Alarm type 6-13
ALARM_TASK 10-97
Alarmhandling in the PLC 10-97
Alarms 6-9
ALIAS 10-37
Alignment at pointer accesses 10-35
Alternative branch 2-21
Alternative Branch in SFC 2-21, 5-40
Analyzation of expressions 10-61
AnalyzationNew.lib 10-61
AND 10-4
AND Operator in AWL 2-9
Append Program Call 6-54
Append Task 6-51, 6-52
Arc cosine 10-23
Arc sine 10-26 10-23
Arc tangent 10-23
archive 10-72
Archive ZIP 4-25
Argument 2-1, 2-3
Arguments 2-5
Arrange Symbols 4-82
Array
 Declaration dialog 5-8
ARRAY 10-33
Array\Access 10-33
Array\Initialization 10-33
Arrays in parameter manager 6-70
AS
 tooltip for step attributes 5-43, 5-44
ASCII format for trace 6-65
asd-file 4-4
ASIN 10-23
Ask for project info 4-5
asl.-file * MERGEFORMAT 4-5
Assignment 2-11, 5-31
Assignment Combs 5-31
Assignment in FBD 5-29
Assignment operator 2-13
AT 5-6
AT Declaration 5-6
ATAN 10-23
Auto Clear Mode 6-30
Auto delete 6-48
Auto Load 4-5
Auto reset 6-48
Auto Save 4-4
Auto save before compile 4-5
Autodeclaration 4-6, 5-8
Autoformat 4-6
Automatic startup 6-43
Automatic Startup 6-30
Available connections
 DeviceNet-Slave 6-49

B

Back one macro level 5-57
Backup automatic 4-4
bak-file 4-4
Base parameter
 DeviceNet-Master 6-45
Base parameters
 Bitchannel 6-29

- CAN Master 6-36
 - CAN module 6-37
 - Channel 6-28
 - DeviceNet-Slave 6-46
 - DP Master 6-30
 - DP slave 6-32
 - I/O Module 6-25
 - Base settings
 - CanDevice 6-42
 - Batch commands 10-69
 - Baud rate
 - CAN Master 6-36
 - Baudrate
 - DeviceNet-Master 6-46
 - Baudrate 6-42
 - BCD Conversion 10-50
 - BCD_TO_INT 10-50
 - binary application file 4-12
 - binary symbol information 4-20
 - Binding of ST operators 2-11
 - Bit addressing 10-28
 - Bit Strobe 6-48
 - Bitaccess 5-13, 10-28
 - Bit-addressed variable 5-27
 - Bit-addressed variables 5-20
 - BITADR 10-13
 - Bitchannel 6-29
 - Bit-Functions 10-50
 - Bitvalues 4-7
 - BLINK 10-57
 - Block 5-37
 - BOOL 10-31
 - BOOL Constants 10-25
 - BOOL_TO Conversions 10-14
 - Boot project 4-5, 4-7, 4-12, 4-70, 4-81, 10-76, 10-94
 - Box 5-30
 - Box with EN input in LD 5-36
 - Breakpoint
 - Delete 5-22
 - Set 5-21
 - Breakpoint 1-1, 2-24, 5-20
 - Breakpoint 5-22
 - Breakpoint Dialog Box 4-72
 - Breakpoint position 4-71
 - Breakpoint Positions in Text Editor 5-21
 - Broadcast 6-3
 - Browser ini-file 6-77
 - Build 4-11, 4-30, 4-68, 10-71
 - Bus identifier 6-42
 - Bus parameters
 - DP master 6-31
 - BusDiag.lib 6-26
 - Bus-Diagnosis 6-26
 - BY 2-14
 - BYTE 10-31
 - byte alignment 10-86
 - BYTE Constants 10-26
 - Byte-addressing mode 10-94
 - Byte-Functions 10-50
- C**
- C Modifier in AWL 2-9
 - CAL 10-14
 - CAL Operator in AWL 2-9
 - CALC 2-9
 - CALCN 2-9
 - Calculate addresses 6-23, 6-30
 - call 10-72
 - Call of a program 2-5
 - Call tree 4-59
 - callback function 6-55
 - Calling a function 2-1
 - Calling a function block 2-4, 2-11
 - Calling function blocks in ST 2-13
 - Calling POUs in text editors 5-20
 - callstack 6-57
 - CAN Master
 - Base parameters 6-36
 - CAN Parameters 6-36
 - Module Parameters 6-37
 - CAN Modular Slave
 - Modules selection 6-39
 - CAN Module
 - Base parameters 6-37
 - CAN Parameters 6-38
 - Modules selection for modular slaves 6-39
 - PDO Mapping 6-39
 - CAN Modules 6-36
 - CAN parameters
 - CAN Master 6-36
 - CAN module 6-38
 - CAN settings
 - CanDevice 6-42
 - CAN Settings 6-3
 - CAN settings for network variables list 6-4
 - CanDevice
 - Base settings 6-42
 - CAN settings 6-42
 - Configuration 6-41
 - Default PDO mapping 6-43
 - CANopen Libraries 6-41
 - CANopen-Node 6-41
 - CANopen-Slave 6-41
 - Cascade 4-82
 - CASE 2-14
 - CASE instruction 2-14
 - CASEFOR loop 2-11
 - CFC
 - Back one/all macro level 5-57
 - Changing connections 5-52
 - Connection marker 5-52
 - Copy elements 5-51
 - Create macro 5-56
 - Creating connections 5-51
 - Cursor positions 5-47
 - Deleting connections 5-52
 - Display order 5-53
 - Edit macro 5-56
 - EN/ENO 5-50
 - Expand macro 5-57
 - Feedback paths 5-57
 - Insert Box 5-48
 - Insert Comment 5-49
 - Insert In-Pin 5-49
 - Insert Input 5-48
 - Insert Input of box 5-49
 - Insert inputs/outputs 5-53
 - Insert Jump 5-48
 - Insert Label 5-49
 - Insert Out-Pin 5-49
 - Insert Output 5-48
 - Insert Return 5-49

- Moving elements 5-51
- Negation 5-49
- Order – One backwards 5-54
- Order – One forwards 5-54
- Order – To the beginning 5-54
- Order – To the end 5-54
- Order according data flow 5-54
- Order of execution 5-53
- Order topologically 5-53
- Properties of POU's 5-50
- Select elements 5-51
- Set/Reset 5-49
- CFC 2-21
- CFC in Online mode 5-57
- cfg-file 6-24
- Change of State 6-48
- Change values online 2-24
- channel 6-28
- Channel
 - Channelparameters 6-29
 - Custom Parameters 6-29
- Channel 6-28
- Channel-Id 6-28
- Channelparameter 6-29
- CHARCURVE 10-59
- check 10-71
- Check at Login 4-8
- Check automatically 4-13
- Check for overlapping addresses 6-23
- Check in 4-19
- Check In 4-49
- Check out 4-18
- Check Out 4-49
- Check product code for DeviceNet-Slave 6-46
- Check product version for DeviceNet-Slave 6-46
- Check project
 - Concurrent Access 4-45
 - Multiple write access on output 4-45
 - Overlapping memory areas 4-44
 - Unused Variables 4-44
- Check project 4-44
- Check vendor id for DeviceNet-Slave 6-46
- CheckBounds 10-34
- CheckDivReal 10-2
- CheckPointer function 10-35
- CheckPointerAligned function 10-35
- CheckRangeSigned 10-37
- CheckRangeUnsigned 10-37
- Checks for DeviceNet-Slave 6-46
- checksum 4-81
- ci-file 4-25
- clean 10-71
- Clean all 4-31
- Close all 4-82
- Close file 4-23
- CMS Priority Group 6-39
- COB-Id 6-36, 6-39
- COB-ID 6-4
- CoDeSys 1-1
- Coil 2-22, 5-36
- Colors 4-9
- Command entry in the PLC-Browser 6-77
- Command file 10-70
- Command Line 10-69
- comment
 - nested 4-11
- Comment
 - I/O Module 6-25
 - Comment 5-1, 5-9, 5-25
 - Comment in CFC 5-49
 - Communication
 - DDE 8-1
 - Symbolic interface 4-15
 - Communication 4-15
 - Communication Cycle Period 6-36
 - Communication Parameters
 - Check at Login 4-8
 - Dialog 4-80, 4-81
 - do not save in project 4-8
 - Quick check 4-81
 - Saving with project 4-8
 - Communication Parameters 4-77
 - Communications timeout 4-8
 - Communications timeout for download 4-8
 - Compare 4-38
 - Compare Project
 - Working in Compare Mode 4-41
 - Compare Project 4-41
 - Compare projects 4-38, 4-39
 - Compare with ENI-Project 4-38
 - Comparing projects 4-38
 - Comparison result 4-39
 - Compilation 4-68
 - compile 10-71
 - Compile File options 4-19
 - Compile files directory 4-10
 - Compiler version 4-12
 - Compress 6-65
 - Compression 10-98
 - CONCAT 10-40
 - Concatenation 10-40
 - Concurrent Access 4-45
 - Configuration file 6-24
 - Configuration files directory 4-10
 - Configuration of CAN modules 6-36
 - Configuration of Profibus Modules 6-29
 - Configure ENI 4-17
 - Configure symbol file 4-15
 - Configured input connections
 - DeviceNet-Slave 6-49
 - Connection marker in CFC 5-52
 - Connections 5-52
 - Connections in CFC 5-51, 5-52
 - CONSTANT 5-5
 - Contact 2-22, 5-35
 - Content Operator 10-14, 10-34
 - Context Menu 4-3
 - Context Sensitive Help 4-83
 - Continuous function chart editor 2-21
 - Continuous Function Chart Editor (CFC) 5-47
 - controlled variable 10-54, 10-55
 - Controller Index 6-4
 - controller status 4-81
 - Conversion of Integral Number Types 10-16
 - Conversions of types 10-14
 - Convert object 4-56
 - Converting of old PLC configurations 6-24
 - Converting S5 to IEC 1131-3 10-80
 - Copy 4-61
 - Copy object 4-56, 10-73
 - Copying elements in CFC 5-51
 - Copying in FBD 5-32
 - COS 10-22
 - Cosine 10-22

Create Backup 4-4
 Create binary file of the application * MERGEFORMAT
 4-12
 Create binary symbol information 4-20
 Create boot project 4-20, 10-76
 Create boot project 4-81
 Create macro in CFC 5-56
 Create translation file 4-32
 CreateBasicSDOs 6-39
 CreateCommSDOs 6-39
 CreateMappingSDOs 6-39
 Cross Reference List 4-59
 CTD 10-46
 CTU 10-45
 CTUD 10-46
 CurrentVisu 10-97
 Cursor positions in FBD 5-28
 Cursor positions in the CFC 5-47
 Cursor Positions in the LD Editor 5-34
 Cursor setting in FBD 5-29
 Custom parameters
 I/O Module 6-28
 Custom Parameters 6-24, 6-29
 Cut 4-61
 Cutting in FBD 5-32
 Cycle independent forcing) 10-94
 Cyclic 6-48
 cyclic task 6-53

D

Data Base Link
 Add Shared Objects 4-52
 Check In 4-49
 Check Out 4-49
 Define 4-48
 Get All Latest Versions 4-51
 Get Latest Version 4-49
 Label Version 4-52
 Login 4-48
 Multiple Check In 4-51
 Multiple Check Out 4-51
 Multiple Define 4-51
 Multiple Undo Check Out 4-51
 Project Version History 4-51
 Refresh Status 4-53
 Show Differences 4-49
 Show Version History 4-49
 Undo Check Out 4-49
 Data Base Link 4-46
 Data type 5-6
 data type name 10-105
 Data types 2-8, 4-2
 database project 4-17
 Datat Control Time 6-31
 DATE 10-32
 DATE Constants 10-25
 Date/Time in alarm log-file 6-16
 DATE_AND_TIME 10-32
 DATE_AND_TIME Constants 10-26
 DATE_TO Conversions 10-18
 DCF file 6-36
 DCF file for creating Global Variables list 6-3
 DCF write 6-38
 DDE 8-1
 DDE inquiry
 General approach to data 8-1
 DDE inquiry 8-1
 DDE Interface
 Accessing variables with Intouch 8-2
 Activate 8-1
 Linking variables using EXCEL 8-2
 Linking variables using WORD 8-1
 Which variables can be read? 8-1
 DDE Interface 8-2
 Deactivate alarm evaluation in online mode 6-16
 Deactivate task generation 10-98
 Deactivation variable 6-13
 Debug Task 6-57
 Debugging 1-1, 2-23, 4-11, 5-20, 5-27
 Debugging in multitasking environment 10-85
 declaration 10-103
 Declaration
 AT 5-6
 automatic 5-8
 New 5-10
 Pragma 5-11
 Declaration 5-3
 Declaration 5-13
 Declaration 5-14
 Declaration Editor
 Line numbers 5-9
 Online Mode 5-10
 Declaration Editor 5-3
 Declaration keyword 5-6
 Declaration of a variable 5-6
 Declaration Part 2-1, 5-3
 Declaration window 5-1
 Declarations as tables 4-6, 5-9
 Declare Variable 4-66
 Decrementer 10-46
 Default value of DeviceNet-Slave parameter 6-49
 default.chk 4-81
 default.prg 4-81
 default.sts 4-81
 Define 4-48
 delay 10-73
 DELETE 4-62, 10-41
 Delete a label 5-42
 Delete Action in SFC 5-42
 Delete Object 4-54
 Delete Step and Transition in SFC 5-40
 Delete Transition 5-42
 Deleting in FBD 5-32
 Demo mode 9-1
 Dereferencing 10-14, 10-34
 DERIVATIVE 10-51
 Desktop 4-8
 Device active in configuration 6-46
 device guid 10-74
 device instance 10-74
 device parameter 10-74
 Device Type 6-42
 DeviceNet
 PLC Configuration 6-45
 DeviceNet-Master
 Base parameter 6-45
 Module Parameters 6-46
 DeviceNet-Master 6-45
 DeviceNet-Slave
 Base parameters 6-46
 I/O connection configuration 6-47
 Module parameters 6-49
 Parameters 6-49

- DeviceNet-Slave 6-45
 - DiagGetBusState 6-27
 - DiagGetState 6-26
 - Diagnosis 6-26, 6-50
 - Diagnosis address 6-26
 - diagnosis messages 6-50
 - Diagnostic address 6-37
 - DINT 10-31
 - DINT Constants 10-26
 - Directory
 - Options 4-10
 - Setting via batch commands 10-72
 - Disable task 6-57
 - Display Flow Control 4-76
 - Display height 10-97
 - Display order in CFC 5-53
 - Display width 10-97
 - DIV 10-2
 - DIV Operator in AWL 2-9
 - DO 2-15
 - Do not save communication parameters in project 4-8
 - Docu File 6-7, 6-8
 - docuframe 4-28
 - Docuframe file 6-7, 6-8
 - Document 4-28, 4-36
 - Document Frame 6-7, 6-8
 - Documentation of the project 4-36
 - Download 4-14, 4-68, 4-70, 4-81
 - Download as file 10-94
 - download information 4-81
 - Download information 4-31, 4-81
 - Download information file 4-70
 - Download of parameter lists 6-74
 - Download of PLC configuration 10-94
 - Download Symbol File 10-94
 - DP Master
 - Base parameters 6-30
 - Bus parameters 6-31
 - DP parameters 6-30
 - DP Master 6-30
 - DP Master 6-30
 - DP parameters
 - DP master 6-30
 - DP slave 6-33
 - DP slave
 - Base parameters 6-32
 - Groups 6-35
 - Input/Output 6-34
 - Module parameters 6-36
 - Properties 6-36
 - DP slave 6-35
 - DP system 6-29
 - Drag&Drop 4-53
 - DSP301 6-36
 - DSP306 6-36
 - DT 10-32
 - DT_TO Conversions 10-18
 - Dump XML symbol table 4-15
 - DWORD 10-31
 - DWORD Constants 10-26
 - Dynamc texts 10-75
 - dynamictextfiles on / off 10-75
 - dynamictexthideelements on / off 10-75
 - dynamictexts on / off 10-75
- ## E
- Edit
 - Autodeclare 4-66
 - Copy 4-61
 - Copy/Paste in CFC 5-51
 - Cut 4-61
 - Cut in FBD 5-32
 - Delete 4-62
 - Find 4-63
 - Find next 4-63
 - Input Assistant 4-64
 - Macros 4-66
 - Next error 4-66
 - Paste 4-62
 - Paste in FBD 5-32
 - Previous error 4-66
 - Redo 4-61
 - Remove library 6-18
 - Replace 4-63
 - Undo 4-60
 - Edit Licensing Information 9-1
 - Edit macro in CFC 5-56
 - Editing functions 4-60
 - Editor
 - Body 5-1
 - Comments 5-1
 - Declaration part 5-1
 - IL 5-23
 - Print margins 5-1
 - Shortcut mode 5-7
 - Syntax Coloring 5-7
 - Editor 5-1
 - Editor for Structured Text 5-24
 - Editor options 4-6
 - Editors 5-1
 - EDS file 6-36, 6-39, 6-41, 6-42, 6-45
 - EDS file for DeviceNet-Slave 6-46
 - EDS file generation 6-42
 - ELSE 2-13, 2-14
 - ELSIF 2-13
 - Emergency Telegram 6-39, 6-43
 - EN Input 2-23, 5-36, 5-37
 - EN POU 2-23
 - EN/ENO in CFC 5-50
 - Enable task 6-57
 - Encrypted CoDeSys project 4-24
 - Encrypted external library 4-24
 - Encrypted internal library 4-24
 - encryption 4-14
 - Encryption 4-24
 - encryption of a project 4-46
 - END_CASE 2-14
 - END_FOR 2-14
 - END_IF 2-13
 - END_REPEAT 2-15
 - END_TYPE 10-35, 10-36, 10-37
 - END_VAR 5-4
 - END_WHILE 2-15
 - ENI 4-16, 4-23, 4-46
 - ENI credentials 4-5
 - ENI data base 7-1
 - ENI parameters 10-75
 - ENI Server 7-1
 - ENI Server Suite 7-1
 - Entry action 2-17, 5-41
 - Entry or exit action 2-17

- Enumeration 10-35
- EQ 10-12
- EQ Operator in AWL 2-9
- error 10-70
- Event for tasks 6-53
- event task 6-53
- Event-Time 6-41
- Exclude object from build 4-12
- Exclude objects 4-12
- Exclude objects from build 4-30
- Execute comparison 4-38
- Exit 4-30
- EXIT 2-11, 2-16
- Exit action 2-17
- EXIT instruction 2-16
- Exit-Action 5-41
- EXP 10-21
- Expand macro in CFC 5-57
- Expected Packet Rate 6-48
- Expert settings for DeviceNet-Slave 6-46
- Exponential Function 10-21
- Exponentiation 10-24
- Export 6-75, 10-71
- Export array entries 4-16
- Export data entries 4-15
- Export file for creating Global Variables list 6-3
- Export module 6-23
- Export project 4-37
- Export structure components 4-15
- Export variables of object 4-15
- Expression 2-11
- EXPT 10-24
- Extended settings for a DeviceNet-Slave 6-48
- Extended settings for DeviceNet-Slave 6-46
- EXTERNAL 5-5
- external event 6-53
- External library 4-23, 6-17
- External trace configuration
 - Load from file 6-63, 6-66
 - Load from PLC 6-63
 - Load from target 6-66
 - Save to file 6-63, 6-66
 - Save to target 6-66
 - Set as project configuration 6-66
- External variable 5-5
- EXTRACT 10-50
- Extras
 - Accept access rights 4-42
 - Accept change 4-41
 - Accept changed item 4-41
 - Accept properties 4-41
 - Add configuration file 6-24
 - Add label to parallel branch 5-41
 - Associate Action 5-45
 - Back one macro level 5-57
 - Calculate addresses 6-24
 - Callstack... 6-57
 - Clear Action/Transition 5-42
 - Compress 6-65
 - Connection marker in CFC 5-52
 - Convert 6-24
 - Create macro in CFC 5-56
 - Cursor Mode 6-64
 - Display order in CFC 5-53
 - Edit macro in CFC 5-56
 - EN/ENO in CFC 5-50
 - Enable / disable task 6-57
 - Expand macro in CFC 5-57
 - External trace configuration 6-63
 - Link Docu File 6-8
 - Load Watch List 6-59
 - Make Docuframe file 6-8
 - Monitoring Active 6-60
 - Monitoring Options 5-20
 - Multi Channel 6-64
 - Negate in CFC 5-49
 - Negate in FBD 5-31
 - Negate in LD 5-38
 - Next difference 4-41
 - Open instance 5-2
 - Open instance in FBD 5-32
 - Options 5-25
 - Options in SFC 5-44
 - Order - One backwards 5-54
 - Order - One forwards 5-54
 - Order - To the beginning 5-54
 - Order - To the end 5-54
 - Order everything according to data flow 5-54
 - Order topologically in CFC 5-53
 - Paste above in LD 5-38
 - Paste after 5-42
 - Paste after in LD 5-38
 - Paste below in LD 5-38
 - Paste Parallel Branch (right) 5-41
 - Previous difference 4-41
 - Properties in CFC 5-50
 - Properties of a library 6-18
 - Read Receipt 6-60
 - Read Trace 6-63
 - Rename Watch List 6-59
 - Replace element 6-22
 - Save Trace 6-65
 - Save trace values 6-65
 - Save Watch List 6-59
 - Select All in CFC 5-51
 - Set Debug Task 6-57
 - Set/Reset in CFC 5-49
 - Set/Reset in FBD 5-31
 - Set/Reset in LD 5-38
 - Settings for alarm configuration 6-16
 - Show grid 6-64
 - Standard configuration 6-24
 - Start Trace 6-62
 - Step Attributes 5-43
 - Stop Trace 6-63
 - Stretch 6-65
 - Time Overview 5-43
 - Use IEC Steps 5-45
 - View in FBD 5-32
 - Write Receipt 6-60
 - Y Scaling 6-64
 - Zoom Action/Transition 5-42
 - Zoom to POU 5-2
 - Zoom to POU in CFC 5-59
- Extras menu
 - Cancel command 6-80
 - History backward 6-80
 - History forward 6-80
 - Print last command 6-80
 - Save history list 6-80

F

F_TRIG 10-44

- F_TRIG in LD 5-37
 - F4 4-8
 - F4 ignores warning 4-8
 - falling edge 10-44
 - Falling edge detection 5-37
 - FBD
 - Assign 5-29
 - Box 5-30
 - Copy 5-32
 - Cursor position 5-28
 - Cut 5-32
 - Delete 5-32
 - Input 5-31
 - Jump 5-29
 - Negation 5-31
 - Output 5-31
 - Paste 5-32
 - Return 5-30
 - set cursor 5-29
 - FBD 2-21
 - FBD Editor 5-28
 - Feedback paths in CFC 5-57
 - Fields 2-1
 - File
 - Close 4-23
 - Exit 4-30
 - New 4-22
 - New form template 4-22
 - Open 4-22
 - Open project from Source Code Manager 4-23
 - Print 4-28
 - Printer Setup 4-28
 - Save 4-23
 - Save as 4-23
 - Save/Mail Archive 4-25
 - file close 10-70
 - file new 4-22, 10-70
 - file open 4-22, 10-70
 - file save 10-70
 - FIND 4-63, 10-42
 - Find next 4-63
 - Flag 5-11
 - float processor 10-85, 10-86, 10-88
 - Flow control
 - FBD 5-33
 - IL 5-23
 - Flow control 4-76
 - Folder 4-53, 4-54
 - Font 4-6
 - fonts 10-97
 - FOR 2-14
 - FOR loop 2-14
 - Force values 4-74, 6-60
 - Forcing 6-60
 - Fragmentation timeout 6-48
 - freewheeling task 6-53
 - Freeze mode 6-30, 6-35
 - FREQ_MEASURE 10-57
 - frequency measurement 10-57
 - Function
 - Insert 5-19
 - Function 2-1
 - function block 5-18
 - Function Block
 - Insert 5-20
 - instance 2-3
 - Function Block 2-3
 - Function Block 2-3
 - Function block call 2-4
 - Function Block Diagram
 - Online Mode 5-33
 - Function Block Diagram (FBD) 2-21
 - Function Block Diagram Editor 5-28
 - Function block in LD 2-23
 - Function block instance address 10-13
 - Function Block Instances 2-3
 - function block name 10-105
 - Function blocks in the Ladder Diagram 2-23
 - Function call 2-1
 - Function declaration 2-1
 - function name 10-105
 - FUNCTION_BLOCK 2-3
 - Functionblock in parameter manager 6-70
- ## G
- Gap Update Factor 6-31
 - Gateway
 - Channel setup 4-79
 - Principle of gateway system 4-77
 - Quick check 4-81
 - Server setup 4-78
 - Gateway 4-77
 - Gateway 10-73
 - GatewayDDE Server
 - Handling 8-2
 - GatewayDDE Server 8-2
 - GatewayDDEServer
 - Command line options 8-4
 - General Approach to Data 8-3
 - Linking variables using WORD 8-4
 - Which variables can be read? 8-3
 - GatewayDDEServer 8-1
 - GE Operator in AWL 2-9
 - GEN 10-57
 - Get All Latest Versions 4-51
 - Get latest version 4-18
 - Get Latest Version 4-49
 - GetBusState 6-25
 - Global constant 5-5
 - Global Constants 6-6
 - Global Replace 4-44
 - Global Retain Variables 6-6
 - Global Search 4-43
 - Global variable list\Object properties 4-57
 - Global variables
 - Constants 6-6
 - Network variables 6-6
 - Objects 6-2
 - Persistent Variables 6-6
 - Remanent Variables 6-6
 - Global variables 6-2
 - Global Variables List
 - Create 6-3
 - Editing 6-5
 - Graphic Editor
 - CFC 5-47
 - FBD 5-28
 - LD 5-33
 - Graphic Editor 5-24
 - Group assignment of a DP slave 6-35
 - Groups
 - DP slave 6-35
 - Groups 6-30

GT 10-11
 GT Operator in AWL 2-9
 Guard Time 6-39

H

Hardware scan 6-50
 Heartbeat 6-39
 Heartbeat Consumer Time 6-43
 Heartbeat Master 6-36
 Heartbeat Producer Time 6-43
 Heartbeat settings 6-43
 Heartbeatrate
 DeviceNet-Slave 6-49
 Help
 Context Sensitive 4-83
 Hitachi SH 10-90
 HYSTERESIS 10-60

I

I/O complete for DeviceNet-Slave 6-48
 I/O connection configuration
 DeviceNet-Slave 6-47
 I/O Module
 Custom parameters 6-28
 I/O ModuleModule parameters 6-28
 identifier
 offline tooltip 5-2
 Identifier 5-6, 10-103
 Identifier for visualization 10-106
 Identnumber 6-33
 IEC 61131-3 2-26
 IEC Step 2-17, 5-45
 IEC steps 2-18
 lecsfc.lib 2-17
 IF 2-13
 IF instruction 2-11, 2-13
 IL
 Online mode 5-23
 IL 2-3, 2-9
 IL Editor 5-23
 IL operator 2-9
 Implicit at load 4-14
 Implicit variables in SFC 2-19
 import 10-71
 Import 4-37, 6-75
 Import from a S5 Project File 10-80
 Import from a SEQ Symbol File 10-79
 Import module 6-23
 Import project 4-37
 Import Siemens files 4-37
 Include Macro library 4-21
 Incrementer 10-45
 Index ranges 10-96
 INDEXOF 10-4
 Infineon C16x 10-87
 Info for DeviceNet-Slave 6-46
 Inhibit Time 6-40
 INI operator 10-24
 Initialization 5-6, 5-11
 Initialization of retain variables 10-24
 Initialization operator 10-24
 Initialization with zero 10-94
 Initialize Inputs 10-94
 Initialize zero 10-94
 In-Pin 5-49

In-Pin' in CFC 5-49
 Input address 6-25, 6-37
 Input and Output Variable 5-4
 Input Assistant
 structured 4-64
 Stuctured Display 4-64
 unstructured 4-64
 Unstructured Display 4-64
 Input Assistant 4-64
 Input handling in visualization 10-98
 Input in CFC 5-48
 Input in FBD 5-31
 Input of box in CFC 5-49
 Input Variable 5-3
 Input/Output of a DP slave 6-34
 Inputbytes of DeviceNet-Slave 6-48
 Inputs
 DeviceNet-Slave 6-49
 Inputs/Outputs configuration
 DeviceNet-Slave 6-47
 Insert
 Add Entry-Action 5-41
 Add Exit-Action 5-41
 Additional Library 6-17
 All Instance Paths 6-7
 Alternative Branch (left) 5-40
 Alternative Branch (right) 5-40
 Append Program Call 6-54
 Append subelement 6-22
 Append Task 6-52
 Assign 5-29
 Box in CFC 5-48
 Box in FBD 5-30
 Box with EN 5-37
 Box with EN in LD 5-36
 Coil 5-36
 Comment 5-25, 5-26
 Comment in CFC 5-49
 Contact 5-35
 Declaration Keyword 5-6
 Falling edge detection 5-37
 Function 5-19
 Function Block 5-20, 5-36
 In-Pin 5-49
 Input in CFC 5-48
 Input in FBD 5-31
 Input of box in CFC 5-49
 Insert at Blocks 5-37
 Insert element 6-22
 Insert Program Call 6-54
 Insert Task 6-52
 Jump in CFC 5-48
 Jump in FBD 5-29
 Jump in LD 5-38
 Jump in SFC 5-41
 Label in CFC 5-49
 Network (after) 5-27
 Network (before) 5-27
 New Declaration 5-10
 New Watch List 6-59
 Operand 5-19
 Operator 5-19
 Out-Pin 5-49
 Output in CFC 5-48
 Output in FBD 5-31
 Parallel Branch (left) 5-41
 Parallel Branch (right) 5-40

- Parallel Contact 5-35
 - Return in CFC 5-49
 - Return in FBD 5-30
 - Return in LD 5-38
 - Rising edge detection 5-37
 - Step Transition (after) 5-40
 - Step Transition (before) 5-40
 - Timer in LD 5-37
 - Transition-Jump 5-41
 - Type 5-6
 - INSERT 10-41
 - Insert declaration 5-9
 - Insert Function 5-19
 - Insert Function Block in text editors 5-20
 - Insert Function in text editors 5-19
 - Insert inputs/outputs in CFC 5-53
 - Insert Operand in text editors 5-19
 - Insert Program Call 6-51, 6-54
 - Insert standard commands 6-77
 - Insert Task 6-51, 6-52
 - Inserting variables 5-2
 - Instance
 - Open 5-2, 5-32
 - Instance 2-3
 - Instance 6-69
 - Instance name 2-3, 2-4
 - Instance Paths for VAR_CONFIG 6-7
 - Instruction 2-9, 2-11
 - Instruction List 2-3
 - Instruction List (IL) 2-9
 - Instruction List Editor 5-23
 - INT 10-31
 - INT Constants 10-26
 - INT_TO_BCD 10-50
 - INTEGRAL 10-52
 - Intel 386 10-85
 - Intel byte order 10-88, 10-90
 - Intel StrongARM 10-88
 - Intellisense Function 5-2
 - Intellisense-function 4-6
 - Internal library 4-23, 6-17
 - Interval
 - DeviceNet-Slave 6-49
 - Interval for tasks 6-53
- J**
- JMP Operator in AWL 2-9
 - Jump 2-21, 5-29
 - Jump in CFC 5-48
 - Jump in LD 5-38
 - Jump in SFC 5-41
 - Jump Label 5-41, 5-42
- K**
- key 4-14, 4-24
 - Key combinations 10-99
 - Keyboard usage for tables 10-75
 - Keyboard usage for tables 10-98
 - Keyboard Use 10-99
 - Keyword 5-5, 5-6
- L**
- Label 5-41
 - Label for networks 5-25
 - Label in CFC 5-49
 - Label Version 4-52
 - Ladder
 - Insert Comment 5-26
 - Ladder Diagram 2-22
 - Ladder Diagram (LD) 2-22
 - Ladder Diagram in Online Mode 5-39
 - Ladder Editor 5-33
 - language 10-75
 - Language
 - 'Show project translated' 4-35
 - 'Toggle translation' 4-36
 - LANGUAGE 6-16
 - Language file 10-74
 - language file on / off 10-74
 - language in visualization 10-75
 - language switching 4-32
 - Languages 2-9
 - LD
 - Cursor Position 5-34
 - Insert at Blocks 5-37
 - Insert Box with EN Input 5-36, 5-37
 - Insert Coil 5-36
 - Insert Contact 5-35
 - Insert Function Block 5-36
 - Insert Jump 5-38
 - Insert Parallel Contact 5-35
 - Insert Return 5-38
 - Paste above 5-38
 - Paste after in LD 5-38
 - Paste below 5-38
 - LD 2-22
 - LD as FBD 2-23
 - LD Editor 5-33
 - LD Operator in AWL 2-9
 - LE 10-11
 - LE Operator in AWL 2-9
 - lecsfc.lib 2-17
 - LEFT 10-39
 - LEN 10-39
 - Length in Bytes 6-49
 - Length of connection for DeviceNet-Slave 6-49
 - Libraries directory 4-10
 - Library
 - AnalyzationNew.lib 10-61
 - Define 6-17
 - External 4-23, 6-17
 - Insert 6-17
 - Internal 4-23, 6-17
 - Licensing 4-25
 - Linking 10-73
 - Properties 6-18
 - Remove 6-18
 - Standard.lib 6-17
 - User defined 6-17
 - Library 2-8
 - Library Declaration Parts 5-13
 - Library Elements 10-63
 - Library Manager
 - Usage 6-17
 - Library Manager 6-16
 - Library path 6-17
 - library private 5-13
 - library public 5-13
 - License info 4-43
 - License information 6-18

- License Management
 - Add license information 9-1
 - Creating a licensed library in CoDeSys 9-1
 - License Management 9-1
 - Licensing information 9-1
 - Licensing libraries 4-25
 - Life Time Factor 6-39
 - LIMIT 10-10
 - LIMITALARM 10-61
 - LIN_TRAFO 10-52
 - Line number field 4-76
 - Line number field in text editors 5-21
 - Line Number in text editors 5-22
 - Line numbers in Declaration Editor 5-9
 - Link Docu File 6-8
 - link pragma 5-13
 - linking POUs 5-13
 - Linking variables using EXCEL 8-4
 - List components 4-6
 - List identifier 6-4
 - List types 6-69
 - LN 10-21
 - Load & Save 4-4
 - Load Download information 4-31
 - Load file from controller 4-82
 - Load module description 6-25
 - Load module state 6-50
 - Load values 6-65
 - Load Watch List 6-59
 - Local Variable 5-4
 - Lock time for sending 6-48
 - Log 6-18
 - LOG
 - Storing 6-20
 - LOG 2-25, 4-10
 - LOG 10-21
 - Log file for project 6-18
 - Log Menu 6-19
 - Logarithm 10-21
 - login 10-71
 - Login 4-68, 10-70
 - Login to Data Base 4-48
 - Login to ENI data base 4-17
 - Logout 4-70, 10-70, 10-71
 - Loop 2-11
 - LREAL 10-31
 - LREAL as REALs 4-12
 - LREAL Constants 10-26
 - LREAL_TO Conversions 10-17
 - LT 10-11
 - LT Operator in AWL 2-9
- M**
- Macro 4-13, 4-20
 - Macro after compile 4-13
 - Macro before compile 4-13
 - Macro in CFC 5-56
 - Macro library 4-21
 - Macros 4-66
 - Macros in PLC-Browser 6-79
 - Main program 2-7
 - Make docuframe file 4-28
 - Make Docuframe file 6-8
 - Managing Projects 4-22
 - manipulated variable 10-54, 10-55
 - Mapping index range 6-44
 - Mappings 6-69, 10-96
 - Mark 4-7
 - Marking blocks in SFC 5-39
 - Master layout 4-57
 - MAX 10-9
 - Max. Retry Limit 6-31
 - Max.Station Delay 6-31
 - Maximum Comment Size 5-25
 - Maximum number of global data segments 10-92
 - Maximum number of POUs 10-92
 - MDI representation 4-8
 - Memory Layout 10-92
 - Memory location 10-30
 - Menu Log 6-19
 - Menu Online 4-66
 - Merge 4-42
 - message file 10-71
 - Message window 4-2, 4-43, 4-82
 - messages 10-71
 - Messages 4-82
 - MID 10-40
 - MIN 10-9
 - Min. Slave Interval 6-31
 - Min./Max. value of DeviceNet-Slave parameter 6-49
 - Min.Station Delay 6-31
 - Minimum Comment Size 5-25
 - MIPS 10-89
 - MOD 10-3
 - Modifier 2-9
 - Modifiers and operators in IL 2-9
 - Modul id 6-37
 - module diagnosis 6-50
 - Module id 6-25
 - Module parameters
 - DeviceNet-Slave 6-49
 - DP master 6-30
 - DP slave 6-36
 - I/O Module 6-28
 - Module Parameters
 - CAN Master 6-37
 - DeviceNet-Master 6-46
 - Module state 6-50
 - Monitoring
 - Declaration editor 5-10
 - Pragma 5-11
 - Text editor 5-20
 - Watch and Receipt Manager 6-59
 - Monitoring 2-24
 - Monitoring Active 6-60
 - Monitoring of complex types 4-7
 - Monitoring Options 5-20
 - More settings for a DeviceNet-Slave 6-48
 - Motorola 68K 10-86
 - MOVE 10-3
 - Moving elements in CFC 5-51
 - MUL 10-2
 - MUL Operator in AWL 2-9
 - Multi Channel 6-64
 - Multicast Poll 6-48
 - Multiple Check In 4-51
 - Multiple Check Out 4-51
 - Multiple Define 4-51
 - Multiple Undo Check Out 4-51
 - Multiple write access on output 4-45
 - multitasking environment 10-85
 - Multi-user operation 7-1
 - MUX 10-10

N

N Modifier in AWL 2-9
 Name of a visualization 10-106
 Naming of identifiers 10-103
 NE 10-12
 NE Operator in AWL 2-9
 Negate in CFC 5-49
 Negation in FBD 5-31
 Negation in LD 5-38
 Nested comments 4-11
 NetVarUdp_lib_V23.lib 6-5
 Network
 Comment 5-25
 Network 5-25
 Network 5-28
 Network (after) 5-27
 Network (before) 5-27
 Network Comments 5-25
 Network editor
 Online mode 5-27
 Network functionality 6-2
 Network in FBD 2-21
 Network in LD 2-22
 Network in SFC 2-16
 Network number 5-25
 Network number field 4-76
 Network type 6-4
 Network variables
 Editing 6-5
 Network variables 6-2, 6-3
 Network variables 6-6
 Network variables 10-96
 Networkfunctionality 10-96
 New declaration 5-9
 New Declaration 5-10
 New from template 4-22
 New Watch List 6-59
 Next error 4-66
 No initialization 6-38
 Node id 6-42
 Node number 6-25
 Nodeguard telegram 6-43
 Nodeguarding 6-39
 Node-Id 6-38
 NodeID 6-36
 nonpersistent 5-18
 NOT 10-6
 Notice at load 4-14
 Number Constants 10-26
 Number of data segments 4-12

O

Object
 Access rights 4-58
 Add 4-54
 Convert 4-56
 Copy 4-56
 Delete 4-54
 DeviceNet-Slave 6-49
 Drag&Drop 4-53
 Folder 4-53, 4-54
 Managing objects in a project 4-53
 Open 4-56
 Properties 4-57
 Rename 4-55

 Tooltip 4-53
 Object 2-1
 Object 4-53
 Object 4-53
 Object Organizer
 Collapse Node 4-54
 Expand Node 4-54
 New Folder 4-54
 Object Organizer 4-2
 Object properties 4-57
 Object template 4-55
 OF 2-14
 offline mode 10-74
 Online
 Breakpoint Dialog Box 4-72
 Communication Parameters 4-77
 Create boot project 4-81
 Display Flow Control 4-76
 Download 4-70
 Force values 4-74
 Load file from controller 4-82
 Log in 4-66
 Logout 4-70
 Release force 4-74
 Reset 4-71
 Reset (cold) 4-71
 Reset (original) 4-71
 Run 4-70
 Show Call Stack 4-76
 Simulation 4-77
 Single Cycle 4-73
 Sourcecode download 4-81
 Step in 4-72
 Step over 4-72
 Stop 4-71
 Toggle Breakpoint 4-71
 Write file to controller 4-81
 Write values 4-73
 Write/Force Dialog 4-75
 Online Change
 Hints 4-68
 ri-file 4-70
 Target setting 10-94
 Online Change 4-67, 4-68
 Online Change on several controllers 4-69
 Online functions 1-1
 Online functions 4-66
 Online in Security mode 4-7
 Online messages from Controller 4-70
 online mode 10-74
 Online Mode
 CFC 5-57
 Declaration Editor 5-10
 FBD 5-33
 LD 5-39
 Network editor 5-27
 PLC Configuration 6-49
 SFC 5-45
 Taskconfiguration 6-55
 Text editor 5-20
 Online settings for Alarm configuration 6-16
 Open a project from PLC 4-22
 Open instance 4-59, 5-2
 Open instance in FBD 5-32
 Open object 4-56
 Open POU 4-59
 Operand 2-1, 5-19

Operator in text editors 5-19
 optimization 10-87
 Optimize 6-31
 Optimized jumps 10-85
 Optimized load operations 10-85
 Optimized operations with constants 10-85
 Optional device 6-38
 Options for Build 4-11
 Options for Colors 4-9
 Options for Directories 4-10
 Options for Editor 4-6
 Options for Load & Save 4-4
 Options for Log 4-10
 Options for 'Macros' 4-20
 Options for project 4-3
 Options for Project objects 4-16
 Options for Project source control 4-16
 Options for 'Symbol Configuration' 4-15
 Options for the Desktop 4-7
 Options for User information 4-5
 OR 10-5
 OR Operator in AWL 2-9
 Order - One backwards in CFC 5-54
 Order - One forwards in CFC 5-54
 Order - To the beginning 5-54
 Order - To the end 5-54
 Order everything according to data flow 5-54
 Order of execution in CFC 5-53
 Order topologically in CFC 5-53
 Original order 5-10
 Out-Pin in CFC 5-49
 Output address 6-25, 6-37
 Output in CFC 5-48
 Output in FBD 5-31
 Output parameters 5-20
 Output Reset in FBD 5-31
 Output Set 5-31
 Output Variable 5-4
 Outputbytes of DeviceNet-Slave 6-48
 Outputs
 DeviceNet-Slave 6-49
 Overlapping memory areas 4-44

P

PACK 10-51
 Pack variables 6-4
 Parallel branch 2-21
 Parallel Branch in SFC 5-41
 Parallel Branch in SFC 2-21, 5-40, 5-41
 Parallel Contact 5-35
 Parallel Contacts 2-22
 Parameter assignment at program call 2-5
 Parameter List
 Download with project 6-73
 Type 6-69
 Parameter Manager
 Array 6-70
 Copy list 6-72
 Cut list 6-72
 Cut/Copy/Paste line 6-73
 Delete line 6-73
 Delete list 6-72
 Export 6-75
 Fade out and fade in lines 6-73
 Format Dec/Hex 6-73
 Function block 6-70
 Import 6-75
 Insert line 6-73
 Insert list 6-71, 6-73, 6-74
 Instance 6-69
 Instance list 6-70
 Line after 6-73
 List types 6-69
 Mappings 6-69
 Monitoring values 6-74
 Online Mode 6-74
 Parameters 6-69
 Paste list 6-72
 Rename List 6-72
 Sorting lists 6-74
 Structure variable 6-70
 System Parameters 6-69
 Template 6-69, 6-70
 Upload and Download 6-74
 Variables 6-69
 Write values 6-74
 Parameter Manager 5-14
 Parameter Manager Editor 6-68
 Parameters
 DeviceNet-Slave 6-49
 Parameter Manager 6-69
 Target Settings 10-96
 Parameters for task 6-54
 Password 4-13, 10-70
 password via command file 10-74
 Passwords 4-46
 Paste above in LD 5-38
 Paste after in LD 5-38
 Paste after in SFC 5-42
 Paste below in LD 5-38
 Paste Parallel Branch 5-41
 Pasting 4-62
 Pasting in FBD 5-32
 P-controller 10-55
 PD 10-54
 PDO 6-39
 PDO mapping 6-43
 PDO mapping of a CAN module 6-39
 PERSISTENT 5-4
 Persistent Global Variables 6-6
 Persistent variable 5-4
 Persistent Variable 6-6
 PI-controller 10-56
 PID 10-55
 PID_FIXCYCLE 10-56
 Placeholders for alarm messages 6-10
 PLC Configuration
 Add configuration file 6-24
 Address check on/off 10-94
 Bitchannel 6-29
 CAN Configuration 6-36
 CanDevice * MERGEFORMAT 6-41
 CANopen-Slave 6-41
 Channel 6-28
 CoDeSys programmable PLC as CANopen-Slave 6-41
 Compatibility 6-20
 Convert old configurations 6-24
 Custom Parameters Dialog 6-24
 DeviceNet 6-45
 Diagnosis 6-50
 Download as file 10-94
 Export module 6-23

- General settings 6-23
- Hardware scan 6-50
- I/O Module 6-25
- Import module 6-23
- Insert/Append elements 6-22
- Online mode 6-49
- Profibus Modules 6-29
- Replacing/switching Elements 6-22
- Selecting 6-22
- Service Data Objects 6-41
- Standard configuration 6-24
- Symbolic names 6-22
- Working in 6-21, 6-22
- PLC Configuration V2.1 6-21
- PLC_PRG 2-7
- PLC-Browser
 - Cancel command 6-80
 - Commands 6-77
 - Function 6-76
 - History 6-80
 - ini-file 6-77
 - Macros 6-79
 - Print last command 6-80
 - Save history list 6-80
- PLC-Browser 6-76
- PLC-Browser 10-94
- PLC-Browser options 6-79
- Pointer
 - Address check 10-35
 - Alignment check 10-35
- Pointer 10-34
- Poll 6-48
- Poll Timeout 6-31
- POU (Program Organization Unit) 1-1, 2-1, 4-2
- POU name 10-105
- Power PC 10-88
- Pragma
 - Bitaccess 5-13
 - Link 5-13
 - nonpersistent 5-18
 - symbol 5-12
- Pragma 5-11
- Pragma 5-13
- Pragma 5-14
- Pragma instruction 5-11
- Pragmas for Library Declaration Parts 5-13
- Pragmas for nonpersistent data types 5-18
- Pragmas for Parameter Manager 5-14
- Preemptive multitasking 10-94
- Previous difference 4-41
- Previous error 4-66
- Print 4-28
- Print margins 5-1
- Print range 4-8
- Printer Setup 4-28
- printersetup 10-71
- prm-file 6-75
- Profibus
 - Addresses 6-30
 - Mode 6-30
- Profibus Channel 6-34
- Profibus Master
 - Bus parameters 6-31
 - DP parameters 6-30
 - Module parameters 6-30
- Profibus Slave
 - Base parameters 6-32
 - DP parameters 6-33
 - Group assignment 6-35
 - Input/Output 6-34
 - Module parameters 6-36
 - Properties 6-36
 - User parameters 6-35
- Program 2-5
- Program call 2-5
- program name 10-105
- Project
 - Add Action 4-59
 - Add object 4-54
 - Build 4-30
 - Check 4-44
 - Clean all 4-31
 - Compare 4-38, 4-39
 - Convert object 4-56
 - Copy object 4-56
 - Data Base Link 4-46
 - Delete Object 4-54
 - Document 4-36
 - Export 4-37
 - Global Replace 4-44
 - Global Search 4-43
 - Import 4-37
 - Load download information 4-31
 - Merge 4-42
 - Object access rights 4-58
 - Open instance 4-59
 - Open object 4-56
 - Options 4-3
 - Project Info 4-42
 - Properties 4-57
 - Rebuild all 4-31
 - Rename object 4-55
 - Show call tree 4-59
 - Show Cross Reference 4-59
 - Siemens Import 4-37
 - Translate into another language 4-31
 - User group passwords 4-46
- Project 1-1, 2-1
- project code 4-81
- project data base 7-1
- Project data base
 - categories 7-2
- Project data base 7-2
- Project data base in CoDeSys
 - Working with 7-2
- Project data base in CoDeSys 7-2
- Project directory 4-10
- Project Info 4-5, 4-42
- Project Objects options 4-17
- Project source control 4-16
- Project version 1.5 4-23
- Project version 2.0 4-23
- Project version 2.1 4-23
- Project Version History 4-51
- Properties in CFC 5-50
- Properties of a DP slave in slave operation of the Profibus 6-36
- Properties of a library 6-18
- Properties of a task 6-53
- Properties of Tools 6-80
- Protection of a project 4-24, 4-46
- PUTBIT 10-51

Q

Qualifier 2-17, 2-18
 query 10-72
 Quiet Time 6-31

R

R Operator in AWL 2-9
 R_TRIG 10-44
 R_TRIG in LD 5-37
 RAMP_INT 10-59
 RAMP_REAL 10-60
 Read Receipt 6-60
 Read Trace 6-63
 REAL 10-31
 REAL Constants 10-26
 REAL_TO Conversions 10-17
 Rebuild all 4-31
 Recalculation of addresses 6-32
 Recalculation of Module addresses 6-24, 6-30
 Receipt Manager 6-57
 Redo 4-61
 Refresh Status 4-53
 Release force 4-74
 Remanent variable 5-4
 Remind of boot project on exit 4-5
 Remove Library 6-18
 Rename Object 4-55
 Rename Watch List 6-59
 REPEAT 2-11, 2-15
 REPEAT loop 2-15
 Replace 4-44, 4-63, 10-41
 Replace constants 4-11
 Replace Element in PLC Configuration 6-22
 Reset 4-71
 Reset (cold) 4-71
 Reset (original) 4-71
 Reset in FBD 5-31
 Reset node 6-38
 Reset output in LD 5-38
 Resources
 Global variables 6-2
 Library Manager 6-16
 Log 6-18
 Network variables 6-2
 Variable Configuration 6-2, 6-6
 Resources 2-8, 4-2, 6-1
 RETAIN 2-3, 5-4
 Retain forcing 10-94
 Retain variable 5-4
 Return 2-11, 2-13, 5-30
 Return in CFC 5-49
 Return in LD 5-38
 RETURN instruction 2-13
 Return to standard configuration 6-24
 Revision control 7-1
 ri-file 4-31, 4-70, 4-81
 RIGHT 10-39
 rising edge 10-44
 Rising edge detection 5-37
 ROL 10-7
 ROR 10-8
 Rotation 10-7, 10-8
 RS 10-43
 RTC 10-49
 Run 4-70, 10-70

S

S Operator in AWL 2-9
 S5 10-80
 Sampling Trace 6-60
 Save (forcing) 10-94
 Save as 4-23
 Save as template 4-55
 Save configuration files in project 6-23
 Save ENI credentials 4-5
 Save file 4-23
 Save Mail/Archive 4-25
 Save trace values
 Values in ASCII file 6-65
 Save trace values 6-65
 Save Watch List 6-59
 Saving alarms 6-15
 Saving with project 4-8
 Scan module configuration 6-50
 Screen divider 4-2
 sdb-file 4-20
 SDO 6-38, 6-41
 SDO creation suppressed 6-39
 Security mode 4-7
 Select configuration directory 6-24
 Select configuration file 6-24
 Selected I/O connection 6-48
 Selecting elements in CFC 5-51
 Selecting of elements in PLC Configuration 6-22
 Sensitivity for task watchdog 6-54
 Sequential Function Chart 2-16
 Sequential Function Chart Editor 5-39
 Sequential Function Chart in Online Mode 5-45
 Service Data Objects 6-41
 Set in FBD 5-31
 set languagefile 10-75
 Set output in LD 5-38
 Set/Reset coils 2-23
 Set/Reset in CFC 5-49
 setreadonly 10-73
 Settings Alarm configuration 6-16
 settings for visualization 10-74
 Settings in PLC Configuration 6-23
 SFC
 Add Entry-Action 5-41
 Add Exit- Action 5-41
 Add Label to parallel branch 5-41
 Alternative Branch (left) 5-40
 Alternative Branch (right) 5-40
 Associate Action 5-45
 Clear Action/Transition 5-42
 Delete jump label 5-42
 Delete Step and Transition 5-40
 Execution of steps 5-45
 IEC Step 5-45
 Jump 5-41
 Marking blocks 5-39
 Online Mode 5-45
 Options 5-44
 Parallel Branch (left) 5-41
 Parallel Branch (right) 5-40
 Paste after 5-42
 Paste Parallel Branch 5-41
 Step Attributes 5-43
 Step Transition (after) 5-40
 Step Transition (before) 5-40
 Time Overview 5-43

- Transition-Jump 5-41
- Zoom Action 5-42
- SFC 2-16
- SFC Editor 5-39
- SFC Flags 2-19
- SFC library 2-17
- SFCActionType 2-19
- SFCCurrentStep 2-19
- SFCEnableLimit 2-19
- SFCError 2-19
- SFCErrorAnalyzation 2-19
- SFCErrorPOU 2-19
- SFCErrorStep 2-19
- SFCInit 2-19
- SFCPause 2-19
- SFCQuitError 2-19
- SFCReset 2-19
- SFCStepType 2-19
- SFCTip 2-19
- SFCTipMode 2-19
- SFCTrans 2-19
- Shared objects 4-16
- Shared Objects options 4-19
- Shift 10-6
- SHL 10-6
- Shortcut in Tools
 - Create new 6-83
- Shortcut in Tools 6-83
- Shortcut Mode 5-7
- Shortcuts of Tools 6-80
- Show Call Stack 4-76
- Show diagnosis messages 6-50
- Show Differences 4-49
- Show grid 6-64
- Show POU symbols 4-7
- Show print area margins 4-8
- Show project translated 4-35
- Show Version History 4-49
- SHR 10-7
- Siemens Import 4-37, 10-79
- Simplified input handling 10-98
- Simulation 2-25, 4-66, 4-77, 10-70
- SIN 10-22
- Sine 10-22
- Single Cycle 2-24, 4-73
- Single step 2-24, 4-72
- Singletask in multitasking 10-94
- SINT 10-31
- SINT Constants 10-26
- SIZEOF 10-4
- Slot time 6-31
- Softmotion 10-94
- Sort by address 5-9
- Sort by name 5-9
- Sort by type 5-9
- Sorting in the Table Editor 5-9
- sorucecodedownload 10-70
- Source control 4-16
- Sourcecode download 4-81
- Sourcedownload 4-14
- SQRT 10-20
- Square Root 10-20
- SR 10-42
- ST 2-11, 5-24
- ST Editor 5-24
- ST operand 2-11
- ST operator 2-11
- ST Operator in AWL 2-9
- Standard commands 6-77
- Standard configuration 6-24
- Standard Function 6-17
- Standard Library 6-17
- Standard POU's 2-1
- Standard.lib 6-17
- Start Trace 6-62
- State 6-50
- Station address 6-30, 6-33
- Statistics 4-42
- STATISTICS_INT 10-53
- STATISTICS_REAL 10-53
- Status bar 4-3, 4-7
- Status of PLC 6-50
- Status scan in SFC 2-19
- Step 2-16
- Step Attributes 5-43
- Step in 4-72
- Step Init 2-17
- step name 5-40
- Step over 4-72
- Step Transition (after) 5-40
- Step Transition (before) 5-40
- Stepping 5-20, 5-27
- stop 10-70
- Stop 4-71
- Stop Trace 6-63
- Store trend data in the PLC 10-97
- Stretch 6-65
- STRING 10-31
- STRING Constants 10-26
- String functions 10-39
- STRING_TO Conversions 10-19
- STRUCT 10-36
- structure 5-18
- Structure 10-36
- Structure variables in parameter manager 6-70
- StructureAccess 10-36
- Structured Text 5-24
- Structured Text (ST) 2-11
- StructureInitialization 10-36
- Structures 2-1
- SUB 10-2
- SUB Operator in AWL 2-9
- Subrange types 10-37
- Supported fonts in the target 10-97
- Suppress monitoring of complex types 4-7
- Symbol configuration 4-15
- Symbol file 4-15, 5-12
- Symbol File download 10-94
- Symbolic interface 4-15
- Symbolic names 6-22
- sym-file 4-20
- Sync mode 6-30
- Sync. Mode 6-35
- Sync.COB-Id 6-36
- Sync.Window Lenght 6-36
- Syncs 6-41
- Syntax Coloring 5-3, 5-7
- SysLibAlarmTrend.lib 10-97
- SysLibTime.lib 6-55
- SysTaskInfo.lib 6-55
- system call 10-74
- System events 6-51
- System Events in the Task Configuration 6-55
- System Flag 10-27

System Parameters 6-69
system state 10-74

T

Table Editor 5-9
tablekeyboardusage_codesys on / off 10-75
tablekeyboardusage_web on / off 10-75
Tab-Width 4-6
TAN 10-22
Tangent 10-22
Target 6-75
Target File 6-75
target id 10-74
Target Rotation Time 6-31
Target Settings
 Dialog 6-76
 General 10-94
 Networkfunctionality 10-96
 Target Platform 10-92
 Visualization 10-97
Target Settings 6-75
Target system
 8051 10-91
 Hitachi SH 10-90
 Intel StrongARM 10-88
 MIPS 10-89
Target system Infineon C16x 10-87
Target system Intel 386 compatible 10-85
Target system Motorola 68K 10-86
Target system TriCore 10-91
Target systems
 Power PC 10-88
Target visualization 10-97
Target-Support-Package 6-75
Task 6-51
Task attributes 6-52
Task Configuration
 Append Task 6-52
 Callstack 6-57
 Execution order 6-57
 Insert Program Call 6-54
 Insert Task 6-52
 Set Debug Task 6-57
 System Events 6-55
 Working in 6-51
Task Configuration 6-51
Task enabling 6-57
task priority 6-52
task properties 6-53
task type 6-53
Taskconfiguration
 in Online Mode 6-55
 status of a task 6-55
 time flow 6-55
Template 4-22, 6-69
Template for EDS file 6-42
Template for objects 4-55
Text editor
 Breakpoint 5-21
Text editor 5-21
Text in visualization 4-32
The Continuous Function Chart Editor (CFC) 2-21
The Standard 2-26
THEN 2-13
Tile Horizontal 4-82
Tile Vertical 4-82

TIME 10-32
TIME Constants 10-25
Time for task watchdog 6-54
Time Management in SFC 5-43
Time scan in SFC 2-19
TIME_OF_DAY 10-32
TIME_OF_DAY Constants 10-25
TIME_TO Conversions 10-17
TIME-Function 10-30
timeout 4-8
Timeout
 DeviceNet-Slave 6-48, 6-49
Timer 10-47
Timer in LD 5-37
tit-file 4-32
tnf-File 6-75
TO 2-14
TO_BOOL Conversions 10-15
TOD 10-32
TOD_TO Conversions 10-17
TOF 10-49
Toggle Breakpoint 4-71
Toggle translation 4-36
TON 10-48
TON in LD 5-37
Tool bar 4-7
Tools
 Creating new Shortcuts 6-83
 Executing Shortcuts 6-84
 Frequently asked questions 6-84
 Object Properties 6-80
 Saving Tool Shortcuts 6-84
 Shortcut 6-80
Tools 6-80
Tooltip
 SFC 5-45
Tooltip 4-53, 5-20, 5-27, 5-33, 5-39
Tooltip for comment 5-1
tooltip for identifiers 5-2
TP 10-47
Trace
 Load from file 6-66
 Load from target 6-66
 Save to file 6-66
 Save to target 6-66
 Save trace values 6-65
 Set as project configuration 6-66
Trace 10-94
Trace Buffer 6-60, 6-63
Trace Configuration
 ASCII file 6-65
 Compress 6-65
 Cursor Mode 6-64
 Load values 6-65
 Multi Channel 6-64
 Read Trace 6-63
 Save Values 6-65
 Selection of Trace variables 6-62
 Show grid 6-64
 Start Trace 6-62
 Stop Trace 6-63
 Stretch 6-65
 Trace Buffer 6-63
 Values in ASCII file 6-65
 Y-Scaling 6-64
Trace Variable 6-62
transformation 10-52

- Transition condition 5-42
- Transition condition 2-17
- Transition in SFC
 - Zoom 5-42
- Transition in SFC 2-17
- transition name 5-40
- Transition to timed out 6-48
- Transition-Jump 5-41
- Translate into another language 4-31
- Translate Project (into another Language) 4-35
- translation file 4-32
- Translation file
 - Creation 4-32
 - Editing 4-34
- Translation file 4-31
- Transmission Type 6-40
- Treat LREAL as REAL 4-12
- Trend data 10-97
- TREND_TASK 10-97
- TriCore 10-91
- Triggervariable 6-15
- TRUNC 10-19
- TSDR 6-33
- TSP 6-75
- Turn-off delay 10-49
- Turn-on delay 10-48
- Typ of DeviceNet-Slave parameter 6-49
- Type 5-6
- TYPE 10-35, 10-36, 10-37
- Type Conversions 10-14
- Typed Literal 5-5

U

- UCMM 6-46
- UDINT 10-31
- UDINT Constants 10-26
- UDP settings for network variables list 6-4
- UINT 10-31
- UINT Constants 10-26
- Undo 4-60
- Undo Check Out 4-49
- UNPACK 10-51
- UNTIL 2-15
- Unused Variables 4-44
- update task 6-42
- Upload files directory 4-10
- Upload of parameter lists 6-74
- Use output bit 6-49
- Use source control (ENI) 4-16
- Use VISU_INPUT_TASK 10-98
- User group 4-45
- User group passwords 4-46
- User information 4-5
- user level 10-74
- User parameters
 - DP slave 6-35
- user password 10-74
- User-defined Libraries 6-17
- USINT 10-31
- USINT Constants 10-26

V

- Value of DeviceNet-Slave parameter 6-49
- VAR 5-4
- VAR PERSISTENT 5-4, 6-6

- VAR RETAIN 5-4, 6-6
- VAR_CONFIG 6-2, 6-6
- VAR_CONSTANT 5-5, 6-6
- VAR_EXTERNAL 5-5
- VAR_GLOBAL 6-2
- VAR_IN_OUT 5-4, 10-94
- VAR_INPUT 5-3
- VAR_INPUT CONSTANT 5-50
- VAR_OUTPUT 5-4
- Variable
 - Insert in Editor 5-2
- Variable 5-2
- Variable Configuration
 - Insert Instance Paths 6-7
- Variable Configuration 6-6
- Variable declaration
 - Pragma 5-13
- Variable declaration 5-11
- Variable declaration 5-14
- Variable name 5-6
- Variables 6-69, 10-96
- Variables declaration 5-6
- VARIANCE 10-53
- Vendor ID 9-1
- View in FBD 5-32
- VISU_INPUT_TASK 10-97
- VISU_TASK 10-97
- visual settings 10-74
- visual webvisuactivation on/off 10-75
- Visualization 10-97
- Visualization name 10-106
- Visualization text 4-32
- Visualization without master layout 4-57

W

- Watch and Receipt Manager
 - Force Values 6-60
 - Insert New Watch List 6-59
 - Load Watch List 6-59
 - Offline Mode 6-58
 - Online Mode 6-59
 - Read Receipt 6-60
 - Rename Watch List 6-59
 - Save Watch List 6-59
 - Write Receipt 6-60
- Watch and Receipt Manager 6-57
- Watch list 6-58
- Watch List 6-57
- Watch Variable 5-10, 5-33
- Watchdog 6-33
- Watchdog for tasks 6-54
- Watchdog time 6-54
- Watchdog Time 6-31
- watchlist 10-73
- webvisuactivation on/off 10-75
- Web-Visualization on/off 10-75
- WHILE 2-15
- WHILE loop 2-11, 2-15
- Window
 - Arrange symbols 4-82
 - Cascade 4-82
 - Close All 4-82
 - Library Manager 6-16
 - Log 6-18
 - Messages 4-82
 - Tile Horizontal 4-82

Tile Vertical 4-82
Window 4-82
Window set up 4-82
With arguments 2-5
WORD 10-31
WORD Constants 10-26
Work space 4-2
Write access 4-16
Write file to controller 4-81
Write protection 4-24, 4-46
Write protection password 4-13
Write Receipt 6-60
Write values 4-73
Write/Force Dialog 4-75

X

XE 1-1

XML-Encoding 4-8
XOR Operator in AWL 2-9

Y

Y Scaling 6-64

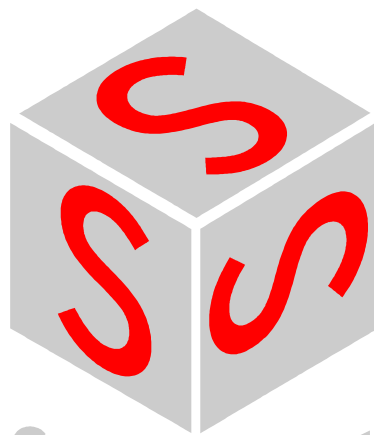
Z

zip-files for Web-Visualization 10-98
Zoom Action in SFC 5-42
Zoom in graphic editors 5-24
Zoom to POU 5-2
Zoom to POU 5-2
Zoom to POU in CFC 5-59
Zoom Transition 5-42

This Page Left Intentionally Blank

First Steps with CoDeSys

Last update: 05.03.2004



S m a r t
Software
Solutions

CONTENT

1	STARTING CODESYS	3
2	WRITING THE FIRST PROGRAM	3
3	A VISUALIZATION FOR THIS	7
4	START THE TARGET SYSTEM	9
5	SETTINGS FOR ESTABLISHING THE CONNECTION	9
6	START THE PROJECT	9
7	CONTINUING FROM HERE	10

1 Starting CoDeSys

Start the CoDeSys programming system with

Start -> Programs -> 3S Software -> CoDeSys V2.3 -> CoDeSys V2.3

2 Writing the First Program

- **Task:**

A running machine is to be observed by a machine operator. Correct running must be confirmed at specified intervals. If there is no confirmation, a warning is first issued, shortly after which the machine will be stopped.

The machine moves an arm along a rectangular path, and completed circuits are to be counted.

- **Creating a New Project**

Starting is easy. Create a new project with the command **File -> New**.

- **Target Settings**

For our first project please select the entry which fits to your runtime system as target in the combo box 'Configuration', e.g. CoDeSys SP for Windows NT Realtime

- **The PLC_PRG POU**

In the **New POU** dialog select the **language of the POU FBD** (Function Block Diagram) and retain the settings **Type Program** and **Name PLC_PRG**.

PLC_PRG is a special POU which is cyclically called and executed by the run-time system in operation.

- **Declaration of the Confirmation Switch**

We will start with the confirmation switch. Mark the three question marks **???** in the first network, and type in the name of the switch (e.g. **Observer**). Now press the right-arrow key. The dialog which appears for the declaration of variables contains the **Name Observer** and the **Type BOOL**. Alter the **Class** to **VAR_GLOBAL** (for global variables). The variable is entered in the object for global variables with **OK**. The following text appears there:

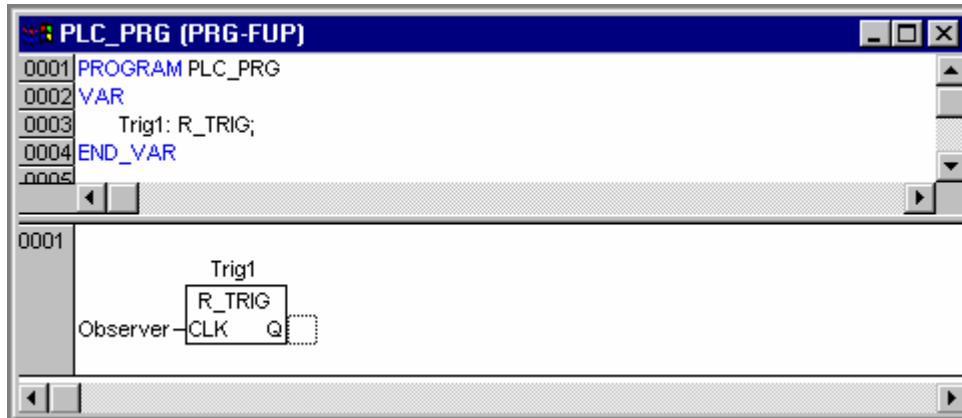
```
VAR_GLOBAL
    Observer: BOOL;
END_VAR
```

- **Rising Edge of the Confirmation Switch**

Everything is OK if the operator is there and confirms correct sequence of operation. That is, if the switch changes from the off state (FALSE) to the on state (TRUE), known as the rising edge, or from on to off, called the falling edge. We will start by determining the rising edge, i.e. whether the switch changes from FALSE to TRUE.

For this, we return to the PLC_PRG POU. Mark the location behind the Observer variable, so that a rectangular marking appears. Execute the **Box** command through the quick menu (right mouse button).

A box for the **AND** operator is inserted and the name is marked. Now call the input assistant with F2. You get a dialog, which offers all available operators. Select first the **Standard Function Blocks** category and there the trigger function block **R_TRIG**, which belongs to the standard.lib. R_TRIG reports the rising edges of boolean value, which in this case is our switch. An instance (data copy) of R_TRIG must be created. To do this, replace the three question marks **???** which appear on top of the inserted box by the instance name (e.g. **Trig1**). After that either click with the left mouse button behind the function block or press the right-arrow key. The **Declare variables dialog** will appear, where the **Class VAR** (for local variables), the **Name Trig1** and the **Type R_TRIG** are already entered. Press **OK** to enter the variable into the declaration segment of the POU.



- **Falling Edge of the Confirmation Switch**

Mark the location behind the function block (rectangular marking), execute the **Box** command via the quick menu, and alter the designation **AND** to **OR** (logical or). Mark the line of the second input of the OR-operator and insert a **Box** of **Type F_TRIG** for the falling edge, having the instance **Name Trig2**.

Mark the three question marks in front of the Trig2 function block, press the <F2> function key (input assistance) and select the **Observer** variable in the **Global Variables** category of the dialog which appears.

- **Time Monitoring, First Part**

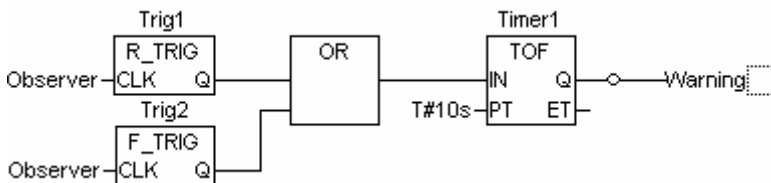
Insert a **Box** of **Type TOF** (switch-off delay) with the **Name Timer1** behind the OR-operator using the quick menu.

Replace the three question marks at the **PT** input with the time constant **T#10s** for 10 seconds. (The time can later be increased, after successful tests.)

- **Issue Warning**

Insert an **Assignment** behind the Q output of the Timer1 function block using the quick menu. Replace the three question marks with the variable name **Warning**. In the declaration of variables, enter the **Class VAR_GLOBAL** and the type **BOOL**.

So that the warning is indeed properly given, mark the first position on the assignment and execute the **Negate** command via the quick menu. This causes the boolean signal to be inverted, and is displayed by means of a small circle.



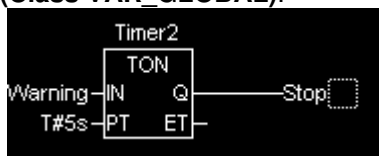
- **Set the Stop Signal After Exceeding the Second Time Limit**

Create another network with the menu command **Insert->Network (after)**.

Use the quick menu to add a **Box** of **Type TON** (function_block, switch-on delay) with the **Name Timer2** into the second network.

Allocate the variable **Warning** to the **IN** input with the aid of the <F2> key, and the time constant **T#5s** to the **PT** input.

After the Timer2 function block we need an **Assignment** again, this time to the variable **Stop** (**Class VAR_GLOBAL**).



- **Insert POU for the Machine**

Insert another POU with the **Name Machine**, of **Type program** and with the **Language SFC** (Sequential Function Chart) into the object organizer (the left hand area of CoDeSys), under

POUs, using the quick menu command **Add object**.

The initial diagram of a POU in SFC consists of a step "Init", and of a subsequent transition "Trans0" followed by a jump back to Init.

- **Specify the Machine's Sequence**

Each phase in the operation of our machine requires a step.

Mark the transition **Trans0** in such a way that a dotted frame appears round Trans0, and use the quick menu to select the command **Step-Transition (after)**. Repeat this procedure five times. If you click directly on the name of a transition or of a step it will be marked in blue and can be altered.

The step after the **Init** step should be called **Go_Right**, and the following steps **Go_Down**, **Go_Left**, **Go_Up** and **Count**.

- **Programming the First Step**

After a double click on the **Go_Right** step you will be asked about the desired **Language**. Select **ST** (structured text) and a separate window appears for the action associated with this step. The machine arm should be moved in the X-direction. The program line reads:
X_pos := X_pos + 1 ;

Complete the line with a return, and declare the variable **X_pos** having **Type INT** for integer.

A small triangle appears in the upper left hand corner of the step to indicate that this step has been programmed.

- **Programming the Next Steps**

Repeat this procedure for the other steps with altered program lines, and as you do so declare the variables **Y_pos** and **Counter** of **Type INT**.

In the **Go_Down** step **Y_pos := Y_pos + 1 ;**

In the **Go_Left** step **X_pos := X_pos - 1 ;**

In the **Go_Up** step **Y_pos := Y_pos - 1 ;**

In the **Count** step **Counter := Counter + 1 ;**

- **Specify Transitions**

Transitions are the conditions for progressing from one phase to the next. Overwrite the first transition after Init with the variable **Start**. In the declaration of variables enter the **Class VAR_GLOBAL** and the **Type BOOL**. The machine thus starts to run when the start switch is pressed.

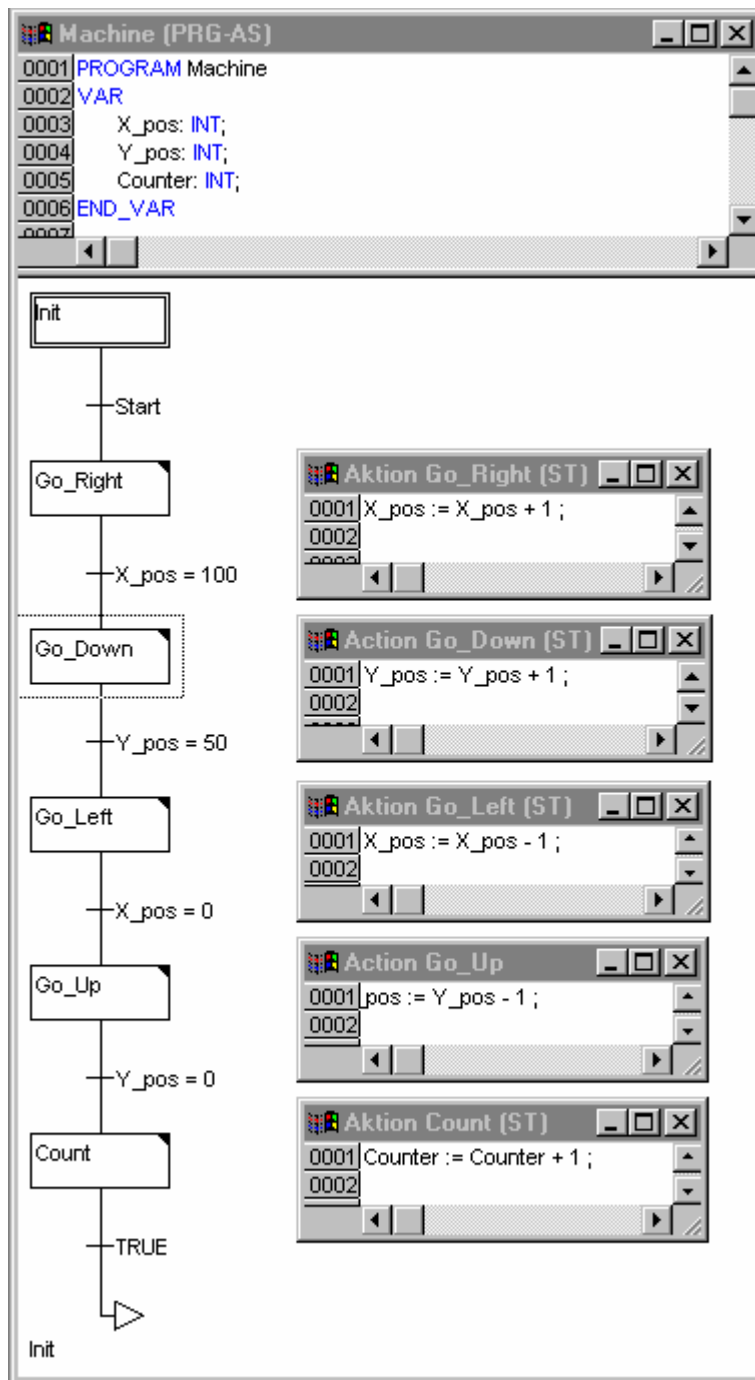
The second transition contains the condition **X_Pos = 100**, so that when the X-position has reached the value 100 the next phase starts.

The third contains the condition **Y_pos = 50**,

the fourth contains the condition **X_pos = 0**,

the fifth contains the condition **Y_pos = 0 and**

the sixth contains the condition **TRUE** (continue immediately after a single execution)



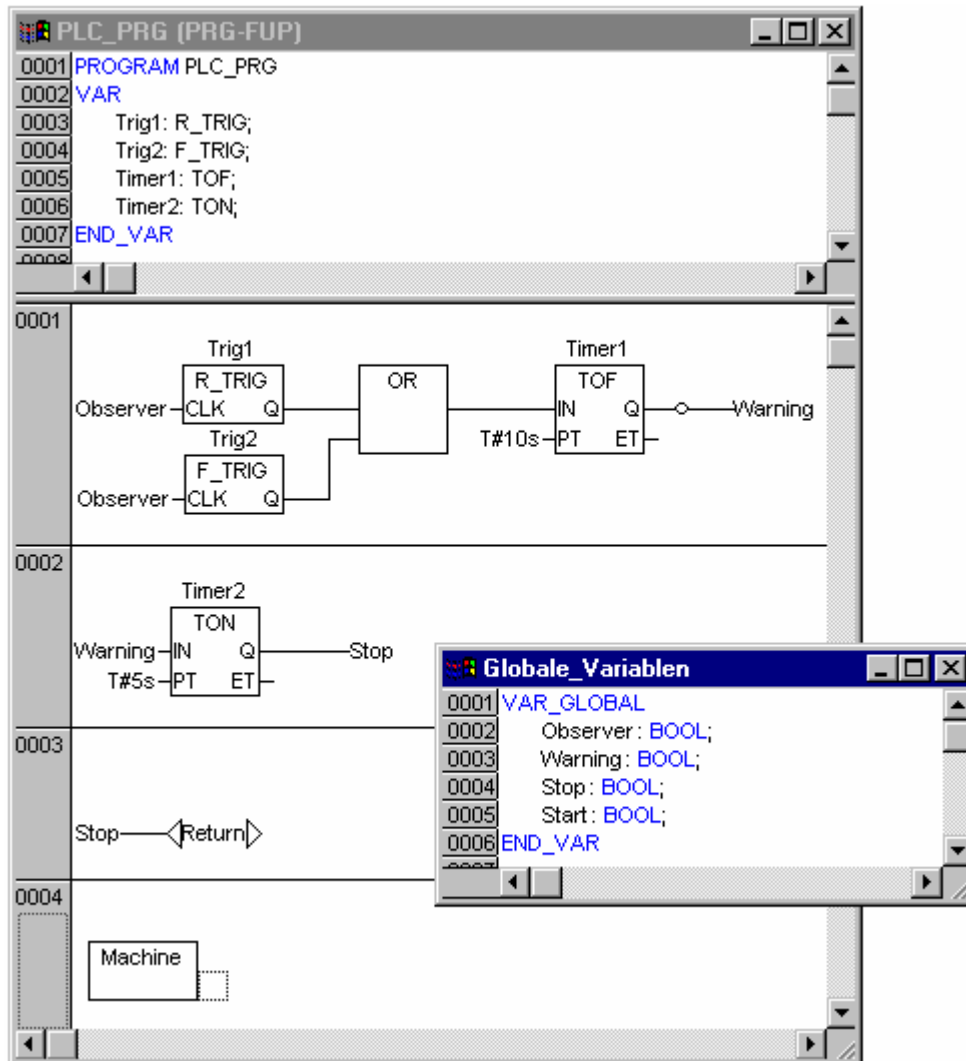
- **Halt Processing at Stop**

Go back to the **PLC_PRG** POU and insert a third network.

Replace the question marks by the variable **Stop**, and after that select the **Return** command from the context menu. The Return has the effect of leaving the PLC_PRG POU when Stop is set.

- **Call Machine POU**

Insert another network, set the focus there and execute in the command **Box** from the context menu. The "AND" in the box is marked already. Press <F2> and in the input assistant dialog which appears, select the **machine** POU in the **User defined Programs category**. This completes the programming.



- **Build Project**

Build the project with the menu command **Project->Rebuild all**, or the <F11> function key. In the message window at the bottom right, „0 errors,, should be reported after the build procedure. If not, check that all the entries have been correctly made. The reported error may also help.

3 A Visualization for This

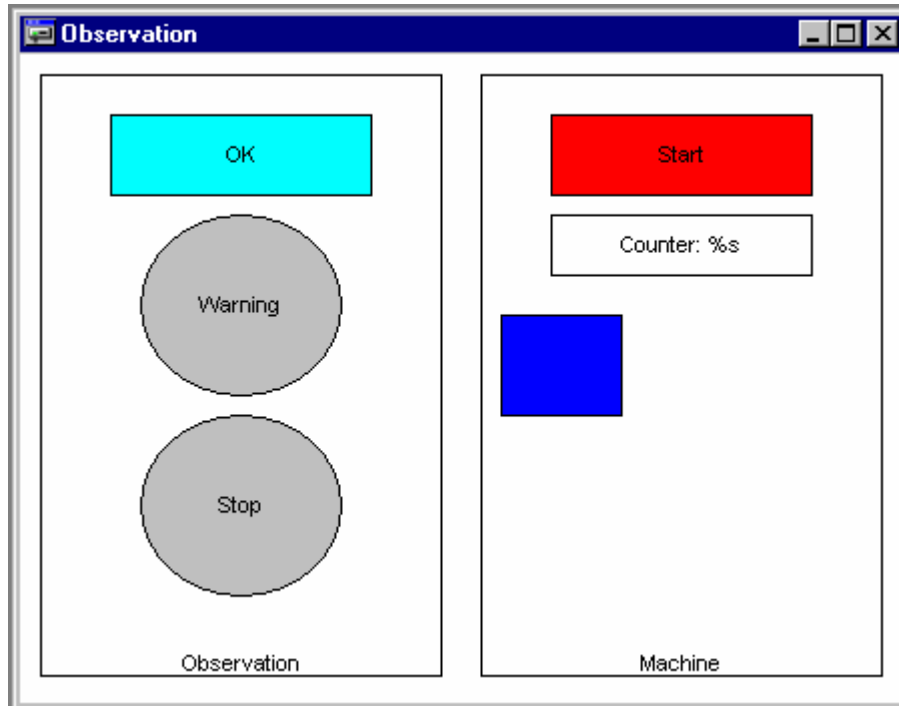
- **Create Visualization**

In the object organizer at the bottom left select the third tab from the left, named Visualization.

- Use the object organizer’s quick menu to call the **Add object** command.

Give the visualization object a **Name**, such as **Observation**.

The visualization should finally look like this:



- **Draw Visualization Element**

Let's start with the confirmation switch (the rectangle with the text OK in the figure). Select the symbol for **Rectangles** from the toolbar.

Press the left hand mouse key in the visualization editor, hold it down, and draw a rectangle with it. Release the mouse key when it has reached the desired form and size.

- **Configure the First Visualization Element**

With a double click on the rectangle, the configuration dialog will appear. In the **Contents** field of the **Text Category** write the text **OK**.

Select the **Variables Category**, click in the **Change Color** field and call up the input assistance with the <F2> function key. Give a double click to the point in the right hand half of the dialog box, and the global variable will appear. Select the Observer variable. The field now contains **.Observer**

After this, select the **Colors Category** in the configuration dialog, and click on the **Inside** button in the **Color** area. Select a color (e.g. light blue). Then select another color with the **Inside** button in the **Alarm color** area (e.g. blue).

In the **Input Category** select the **Toggle variable** point and again enter the variable Observer into the field behind it with <F2>. Leave the configuration dialog.

This causes the rectangle to be colored light blue during running operation when the Observer variable is in the FALSE state, and blue when it is in the TRUE state. With each mouse click the state is changed from TRUE to FALSE, or the other way around.

- **Expand Visualization**

Draw a circle for the warning, and configure it with the following values:

Text Category, Contents field with **Warning**

Variable Category, Color change field with **.Warning**

Colors Category, Color area using **Inside** with color grey, and **Alarm color** with red.

Copy the circle you have created using the menu command **Edit -> Copy** and insert another one with the menu command **Edit -> Paste**. Alter the following values in the new circle:

Text Category, Contents field with the new entry **Stop**

Variable Category, Color change field with the new entry **.Stop**

Draw a rectangle for the start of the machine and configure it with the following values:

Text Category, Contents field with **Start**

Variable Category, Color change field with **.Start**
Input Category, select the point **Toggle variable**
Input Category, behind **Toggle variable**, enter **.Start** in the field
Colors Category, Color area using **Inside** with color red, and **Alarm color** with green.

Draw a rectangle for the counter and configure it with the following values:

Text Category, Contents field with **Counter : %s**
(%s functions as a place holder for the variable value)
Variable Category, Textdisplay field with **Machine.Counter**

Draw a rectangle for the machine and configure it with the following values:

Absolute movement Category, X-Offset field with **Machine.X_pos**
Absolute movement Category, Y-Offset field with **Machine.Y_pos**
Colors Category, Color area using **Inside** with color blue.

If you like, you can draw two large rectangles around the two areas for observation and for the machine, enter **Observation** (or **Machine**) as the **Contents** in the **Text Category**, and select the **Vertical** alignment **bottom**. Using the quick menu command **Send to back** place the rectangle behind the existing elements.

The following steps 4,5,6 only can be executed, if a runtime system has been installed, which fits to the target settings in CoDeSys. Otherwise you have the possibility to work in simulation mode. In this case continue now with Step 6 for logging in and starting the project.

4 Start the Target System

Start the target runtime system. (Please regard that in case of using the runtime system CoDeSys SP RTE this is only possible on Windows NT 4.0, Windows 2000 or Windows XP. Here you will get the icon of CoDeSys SP RTE in the status bar. Click on this with the right mouse button and choose the command 'Start System'.)

5 Settings for Establishing the Connection

- A small number of settings must be made before we can establish the connection to the target computer.
For this purpose carry out the menu command **Online -> Communication parameters**.
- Select the button **New** to configure the connection to your target system. Enter an appropriate name for the connection at the new dialog and select a connection; in case of CoDeSys SP RTE **Shared memory (Kernel)**.
If the target computer is your local computer you can now close the dialog with **OK**. If the target computer is not your local computer, you have to change the value 'localhost' to the Internet address or the target name of the target computer. After that also close the communications parameters dialog with **OK**.

6 Start the Project

- The connection to the target computer will be established using the menu command **Online -> Login** from the **CoDeSys** development environment, and
- with **Online -> Run** your program will be started on the target computer. (If you want to work in simulation mode, the option 'Online' 'Simulation' must be activated).
- By means of the visualization you can now start the machine and operate the confirmation switch.

7 **Continuing From Here**

- It is now up to you to create and use further applications. For further information please see the Online Help and the Users Manual for the CoDeSys Programming System. We wish you every success!

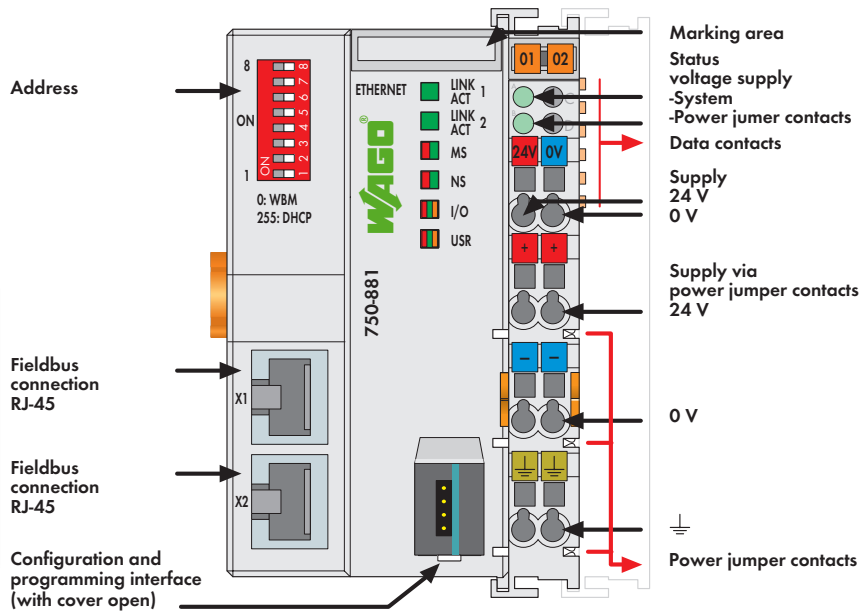
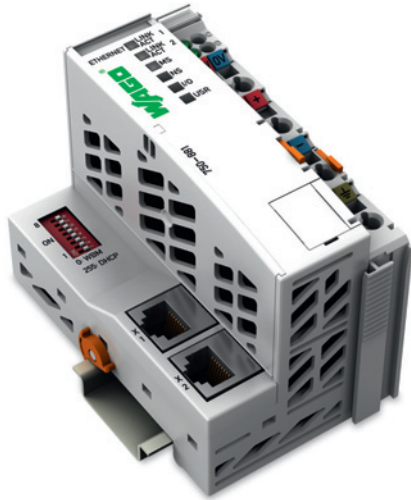
This Page Left Intentionally Blank

WAGO SYSTEM 750-881
ETHERNET
PROGRAMMABLE FIELDBUS
CONTROLLER

Operations & Maintenance Manual
December 2015

PLC - ETHERNET Programmable Fieldbus Controller

32-bit CPU, multitasking



The 750-881 ETHERNET PLC connects ETHERNET to the modular WAGO-I/O-SYSTEM.

The PLC automatically configures, creating a local process image which may include analog, digital or specialty modules. Analog and specialty module data is sent via words and/or bytes; digital data is sent bit by bit.

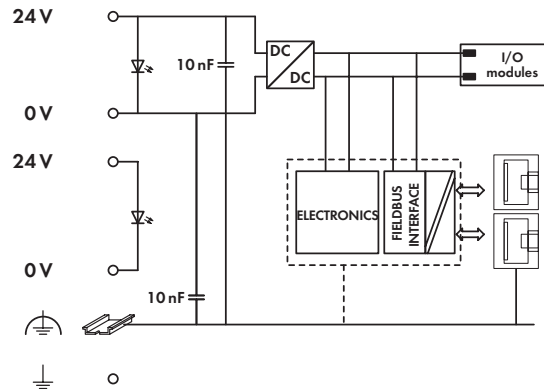
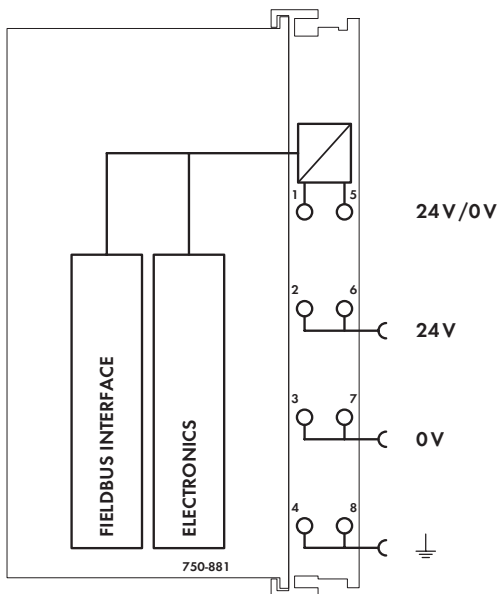
Two ETHERNET interfaces and an integrated switch allow the fieldbus to be wired in a line topology. This eliminates additional network devices, such as switches or hubs. Both interfaces support Auto-Negotiation and Auto-MD(X).

The DIP switch configures the last byte of the IP address and may be used for IP address assignment.

The PLC is designed for fieldbus communication in both EtherNet/IP and MODBUS networks. It also supports a wide variety of standard ETHERNET protocols (e.g., HTTP, BootP, DHCP, DNS, SNTP, SNMP, FTP). An integrated Web server provides the user with configuration options and status information from the controller. The IEC 61131-3 programmable controller is multitasking-capable and features a battery-backed RTC.

Description	Item No.	Pack. Unit
ETHERNET Controller	750-881	1
Accessories		
WAGO-I/O-PRO V2.3, RS-232 kit	759-333	1
Miniature WSB Quick marking system		
plain	248-501	5
with marking	see pages 352 ... 353	
Approvals		
Conformity marking	Also see "Approvals Overview" in Section 1	
Shipbuilding	CE	
UL 508	ABS, DNV, GL, KR	

System Data	
No. of controllers connected to Master	limited by ETHERNET specification
Transmission medium	Twisted Pair S-UTP
	100 Ω, Cat 5;
	Max. line length: 100 m
Baud rate	10/100 Mbit/s
Transmission performance	Class D acc. to EN 50173
Buscoupler connection	2 x RJ-45
Protocols	EtherNet/IP, MODBUS/TCP (UDP), HTTP, BootP, DHCP, DNS, SNTP, FTP, SNMP
Programming	WAGO-I/O-PRO V2.3
IEC 61131-3	IL, LD, FBD, ST, FC



Technical Data

Number of I/O modules	64
with bus extension	250
Fieldbus	
Max. input process image	1020 words
Max. output process image	1020 words
Configuration	via PC
Program memory	1024 Kbytes
Data memory	512 Kbytes
Non-volatile memory (retain)	32 Kbytes
Power supply	24 V DC (-25 % ... +30 %)
Input current typ. at rated load (24 V)	500 mA
Efficiency of the power supply (typ.) at nominal load (24 V)	90 %
Internal current consumption (5 V)	450 mA
Total current for I/O modules (5 V)	1700 mA
Isolation	500 V system/supply

General Specifications

Operating temperature	0 °C ... +55 °C
Wire connection	CAGE CLAMP®
Cross sections	0.08 mm ² ... 2.5 mm ² / AWG 28 ... 14
Stripped lengths	8 ... 9 mm / 0.33 in
Dimensions (mm) W x H x L	62 x 65 x 100
	Height from upper-edge of DIN 35 rail
Weight	161 g
Storage temperature	-25 °C ... +85 °C
Relative air humidity (no condensation)	95 %
Vibration resistance	acc. to IEC 60068-2-6
Shock resistance	acc. to IEC 60068-2-27
Degree of protection	IP20
EMC: CE - immunity to interference	acc. to EN 61000-6-2 (2005)
EMC: CE - emission of interference	acc. to EN 61000-6-3 (2007)
EMC: marine applications	
- immunity to interference	acc. to Germanischer Lloyd (2003)
EMC: marine applications	
- emission of interference	acc. to Germanischer Lloyd (2003)

This Page Left Intentionally Blank



WAGO-I/O-SYSTEM 750 **Programmable Fieldbus Controller ETHERNET** **750-881** **10/100 Mbit/s; digital and analog Signals**

Version 1.1.3



© 2012 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

1	Notes about this Documentation.....	12
1.1	Validity of this Documentation.....	12
1.2	Copyright.....	12
1.3	Symbols.....	13
1.4	Number Notation.....	15
1.5	Font Conventions	15
2	Important Notes	16
2.1	Legal Bases	16
2.1.1	Subject to Changes	16
2.1.2	Personnel Qualifications.....	16
2.1.3	Use of the 750 Series in Compliance with Underlying Provisions	16
2.1.4	Technical Condition of Specified Devices.....	17
2.2	Safety Advice (Precautions).....	18
2.3	Special Use Conditions for ETHERNET Devices.....	20
3	System Description.....	21
3.1	Manufacturing Number.....	22
3.2	Hardware Address (MAC ID).....	22
3.3	Component Update.....	23
3.4	Storage, Assembly and Transport.....	23
3.5	Assembly Guidelines/Standards.....	23
3.6	Power Supply	24
3.6.1	Isolation	24
3.6.2	System Supply	25
3.6.2.1	Connection.....	25
3.6.2.2	Dimensioning.....	26
3.6.3	Field Supply.....	29
3.6.3.1	Connection.....	29
3.6.3.2	Fusing	30
3.6.4	Supplementary Power Supply Regulations	34
3.6.5	Supply Example.....	35
3.6.6	Power Supply Unit	37
3.7	Grounding	38
3.7.1	Grounding the DIN Rail	38
3.7.1.1	Framework Assembly.....	38
3.7.1.2	Insulated Assembly.....	38
3.7.2	Grounding Function.....	39
3.7.3	Grounding Protection	40
3.8	Shielding	41
3.8.1	General	41
3.8.2	Bus cables.....	41
3.8.3	Signal lines	41
3.8.4	WAGO Shield Connecting System	42

4	Device Description	43
4.1	View	46
4.2	Connectors.....	48
4.2.1	Device Supply	48
4.2.2	Fieldbus Connection.....	49
4.3	Display Elements	50
4.4	Operating Elements.....	51
4.4.1	Service Interface.....	51
4.4.2	Mode Selector Switch.....	52
4.4.3	Address Selection Switch.....	54
4.5	Technical Data	55
4.5.1	Device Data	55
4.5.2	System Data.....	55
4.5.3	Supply.....	56
4.5.4	Fieldbus MODBUS/TCP.....	56
4.5.5	Accessories	56
4.5.6	Wire Connection.....	56
4.5.7	Climatic environmental conditions.....	57
4.5.8	Mechanical strength.....	57
4.6	Approvals	58
4.7	Standards and Guidelines	59
5	Assembly	60
5.1	Installation Position.....	60
5.2	Total Extension	60
5.3	Assembly onto Carrier Rail.....	62
5.3.1	Carrier Rail Properties.....	62
5.3.2	WAGO DIN Rail.....	63
5.4	Spacing.....	63
5.5	Assembly Sequence	64
5.6	Inserting and Removing Devices	65
5.6.1	Inserting the Fieldbus Coupler/Controller.....	66
5.6.2	Removing the Fieldbus Coupler/Controller	66
5.6.3	Inserting I/O Module	67
5.6.4	Removing the I/O Module.....	68
6	Connect Devices	69
6.1	Data Contacts/Internal Bus	69
6.2	Power Contacts/Field Supply.....	70
6.3	Connecting a conductor to the CAGE CLAMP®	71
7	Function Description	72
7.1	Operating System	72
7.1.1	Run-up	72
7.1.2	PFC Cycle.....	72
7.2	Process Data Architecture	74
7.2.1	Basic Structure.....	74
7.2.2	Example of an Input Process Image.....	76
7.2.3	Example of an Output Data Process Image.....	77
7.2.4	Process Data MODBUS/TCP and EtherNet/IP	78
7.3	Data Exchange	79

7.3.1	Memory Areas	81
7.3.2	Addressing	84
7.3.2.1	Addressing of I/O Modules	85
7.3.2.2	Address Ranges	86
7.3.2.3	Absolute Addressing	89
7.3.3	Data Exchange between MODBUS/TCP Master and I/O Modules...	91
7.3.3.1	Data Exchange between EtherNet/IP Master and I/O Modules	93
7.3.4	Data Exchange between PLC Function (CPU) and I/O Modules	94
7.3.5	Data Exchange between Master and PLC Function (CPU).....	95
7.3.5.1	Example of MODBUS/TCP Master and PLC Function (CPU)	95
7.3.6	Application Example	97
8	Commissioning	98
8.1	Connecting Client PC and Fieldbus Nodes	99
8.2	Allocating the IP Address to the Fieldbus Node	99
8.2.1	Assigning IP Address via Address Selection Switch	100
8.2.2	Assigning IP Address via DHCP	102
8.2.2.1	Enable DHCP	102
8.2.2.2	Disabling DHCP	103
8.2.3	Assigning the IP Address with a BootP Server	105
8.2.3.1	Note MAC ID	106
8.2.3.2	Determining IP addresses	107
8.2.3.3	Assigning the IP address and Enable BootP	108
8.2.3.4	Disabling BootP	108
8.2.3.5	Reasons for Failed IP Address Assignment.....	111
8.3	Testing the Function of the Fieldbus Node	112
8.4	Preparing the Flash File System	113
8.5	Synchronizing the Real-Time Clock	115
8.6	Restoring Factory Settings	118
9	Programming the PFC using WAGO-I/O-PRO	119
9.1	Configuration using the WAGO-I/O-PRO I/O Configurator	121
9.1.1	Configuration using the "EA-config.xml" File	124
9.2	ETHERNET Libraries for WAGO-I/O-PRO	126
9.3	Functional Restrictions and Limits	128
9.4	General Information about IEC Tasks	131
9.4.1	IEC Task Sequence	133
9.4.2	Overview of Most Important Task Priorities	133
9.5	System Events	135
9.5.1	Enabling/disabling system events	135
9.6	Transfer the IEC program to the controller	137
9.6.1	Transfer via Serial Service Port	138
9.6.2	Transfer via Fieldbus and ETHERNET	141
10	Configuring via the Web-Based Management System (WBM)	143
10.1	Information	144
10.2	Ethernet	146
10.3	TCP/IP	149
10.4	Port	151
10.5	SNMP	153
10.5.1	SNMP V1/V2c	154

10.5.2	SNMP V3	156
10.6	Watchdog	158
10.7	Clock	160
10.8	Security	162
10.9	PLC Info	165
10.10	PLC	166
10.11	Features	169
10.12	I/O Config	170
10.13	WebVisu	172
11	Diagnostics	174
11.1	LED Signaling	174
11.1.1	Evaluating Fieldbus Status	174
11.1.2	Evaluating Node Status - I/O LED (Blink Code Table)	176
11.1.2.1	USR LED	184
11.1.3	Evaluating Power Supply Status	184
11.2	Fault Behavior	185
11.2.1	Loss of Fieldbus	185
11.2.2	Internal Data Bus Failure	187
12	Fieldbus Communication	188
12.1	Implemented Protocols	188
12.1.1	Communication Protocols	188
12.1.1.1	IP (Internet Protocol)	188
12.1.1.2	TCP (Transmission Control Protocol)	193
12.1.1.3	UDP (User Datagram Protocol)	193
12.1.2	Configuration and Diagnostics Protocols	194
12.1.2.1	BootP (Bootstrap Protocol)	194
12.1.2.2	DHCP (Dynamic Host Configuration Protocol)	196
12.1.2.3	HTTP (Hypertext Transfer Protocol)	198
12.1.2.4	DNS (Domain Name Systems)	199
12.1.2.5	SNTP-Client (Simple Network Time Protocol)	199
12.1.2.6	FTP-Server (File Transfer Protocol)	199
12.1.2.7	SNMP (Simple Network Management Protocol)	200
12.1.2.7.1	MIB II Description	201
12.1.2.7.2	Traps	202
12.1.3	Application Protocols	203
12.2	MODBUS Functions	204
12.2.1	General	204
12.2.2	Use of the MODBUS Functions	207
12.2.3	Description of the MODBUS Functions	208
12.2.3.1	Function Code FC1 (Read Coils)	209
12.2.3.2	Function Code FC2 (Read Input Discretes)	211
12.2.3.3	Function Code FC3 (Read Multiple Registers)	213
12.2.3.4	Function Code FC4 (Read Input Registers)	214
12.2.3.5	Function Code FC5 (Write Coil)	215
12.2.3.6	Function Code FC6 (Write Single Register)	216
12.2.3.7	Function Code FC11 (Get Comm Event Counter)	217
12.2.3.8	Function Code FC15 (Force Multiple Coils)	218
12.2.3.9	Function Code FC16 (Write Multiple Registers)	220
12.2.3.10	Function Code FC22 (Mask Write Register)	221

12.2.3.11	Function Code FC23 (Read/Write Multiple Registers)	222
12.2.4	MODBUS Register Mapping	224
12.2.5	MODBUS Registers	227
12.2.5.1	Accessing Register Values	228
12.2.5.2	Watchdog Registers	228
12.2.5.3	Diagnostic Registers	234
12.2.5.4	Configuration Registers	235
12.2.5.5	Firmware Information Registers	240
12.2.5.6	Constant Registers	242
12.3	EtherNet/IP (Ethernet/Industrial Protocol)	244
12.3.1	General	244
12.3.2	Protocol overview in the OSI model	245
12.3.3	Characteristics of the EtherNet/IP Protocol Software	246
12.3.4	EDS File	246
12.3.5	Object Model	247
12.3.5.1	General	247
12.3.5.2	Class Overview	248
12.3.5.3	Explanation of the Table Headings in the Object Descriptions... ..	250
12.3.5.4	Identity (01 _{hex})	250
12.3.5.5	Message Router (02 _{hex})	252
12.3.5.6	Assembly Object (04 _{hex})	253
12.3.5.7	Connection (05 _{hex})	257
12.3.5.8	Connection Manager (06 _{hex})	257
12.3.5.9	Port Class (F4 _{hex})	258
12.3.5.10	TCP/IP Interface (F5 _{hex})	260
12.3.5.11	Ethernet Link (F6 _{hex})	261
12.3.5.12	Coupler/Controller Configuration (64 _{hex})	267
12.3.5.13	Discrete Input Point (65 _{hex})	270
12.3.5.14	Discrete Input Point Extended 1 (69 _{hex})	270
12.3.5.15	Discrete Input Point Extended 2 (6D _{hex})	271
12.3.5.16	Discrete Input Point Extended 3 (71 _{hex})	272
12.3.5.17	Discrete Output Point (66 _{hex})	272
12.3.5.18	Discrete Output Point Extended 1 (6A _{hex})	273
12.3.5.19	Discrete Output Point Extended 2 (6E _{hex})	274
12.3.5.20	Discrete Output Point Extended 3 (72 _{hex})	274
12.3.5.21	Analog Input Point (67 _{hex})	275
12.3.5.22	Analog Input Point Extended 1 (6B _{hex})	276
12.3.5.23	Analog Input Point Extended 2 (6F _{hex})	276
12.3.5.24	Analog Input Point Extended 3 (73 _{hex})	277
12.3.5.25	Analog Output Point (68 _{hex})	278
12.3.5.26	Analog Output Point Extended 1 (6C _{hex})	278
12.3.5.27	Analog Output Point Extended 2 (70 _{hex})	279
12.3.5.28	Analog Output Point Extended 3 (74 _{hex})	280
12.3.5.29	Module Configuration (80 _{hex})	281
12.3.5.30	Module Configuration Extended (81 _{hex})	281
12.3.5.31	Input Fieldbus Variable USINT (A0 _{hex})	283
12.3.5.32	Input Fieldbus Variable USINT Extended 1 (A1 _{hex})	284
12.3.5.33	Input Fieldbus Variable USINT Extended 2 (A2 _{hex})	285
12.3.5.34	Output Fieldbus Variable USINT (A3 _{hex})	286
12.3.5.35	Output Fieldbus Variable USINT Extended 1 (A4 _{hex})	287

12.3.5.36	Output Fieldbus Variable USINT Extended 2 (A5 _{hex})	288
12.3.5.37	Input Fieldbus Variable UINT (A6 _{hex})	289
12.3.5.38	Input Fieldbus Variable UINT Extended 1 (A7 _{hex})	290
12.3.5.39	Output Fieldbus Variable UINT (A8 _{hex}).....	291
12.3.5.40	Output Fieldbus Variable UINT Extended 1 (A9 _{hex}).....	292
12.3.5.41	Input Fieldbus Variable UDINT (AA _{hex}).....	293
12.3.5.42	Input Fieldbus Variable UDINT Offset (AB _{hex}).....	294
12.3.5.43	Output Fieldbus Variable UDINT (AC _{hex})	295
12.3.5.44	Output Fieldbus Variable UDINT Offset (AD _{hex})	296
13	I/O Modules	297
13.1	Overview	297
13.2	Process Data Architecture for MODBUS/TCP	298
13.2.1	Digital Input Modules.....	299
13.2.1.1	1 Channel Digital Input Module with Diagnostics	299
13.2.1.2	2 Channel Digital Input Modules	299
13.2.1.3	2 Channel Digital Input Module with Diagnostics	299
13.2.1.4	2 Channel Digital Input Module with Diagnostics and Output Process Data.....	300
13.2.1.5	4 Channel Digital Input Modules	300
13.2.1.6	8 Channel Digital Input Modules	300
13.2.1.7	16 Channel Digital Input Modules	301
13.2.2	Digital Output Modules.....	302
13.2.2.1	1 Channel Digital Output Module with Input Process Data	302
13.2.2.2	2 Channel Digital Output Modules.....	302
13.2.2.3	2 Channel Digital Input Modules with Diagnostics and Input Process Data.....	303
13.2.2.4	4 Channel Digital Output Modules.....	304
13.2.2.5	4 Channel Digital Output Modules with Diagnostics and Input Process Data.....	304
13.2.2.6	8 Channel Digital Output Module	304
13.2.2.7	8 Channel Digital Output Modules with Diagnostics and Input Process Data.....	305
13.2.2.8	16 Channel Digital Output Modules.....	305
13.2.2.9	8 Channel Digital Input/Output Modules	306
13.2.3	Analog Input Modules.....	307
13.2.3.1	1 Channel Analog Input Modules.....	307
13.2.3.2	2 Channel Analog Input Modules.....	307
13.2.3.3	4 Channel Analog Input Modules.....	308
13.2.4	Analog Output Modules	309
13.2.4.1	2 Channel Analog Output Modules	309
13.2.4.2	4 Channel Analog Output Modules	309
13.2.5	Specialty Modules	310
13.2.5.1	Counter Modules	310
13.2.5.2	Pulse Width Modules.....	312
13.2.5.3	Serial Interface Modules with alternative Data Format.....	312
13.2.5.4	Serial Interface Modules with Standard Data Format	313
13.2.5.5	Data Exchange Module.....	313
13.2.5.6	SSI Transmitter Interface Modules.....	313
13.2.5.7	Incremental Encoder Interface Modules.....	314
13.2.5.8	DC-Drive Controller.....	316

13.2.5.9	Stepper Controller.....	317
13.2.5.10	RTC Module	318
13.2.5.11	DALI/DSI Master Module.....	318
13.2.5.12	EnOcean Radio Receiver.....	319
13.2.5.13	MP Bus Master Module.....	319
13.2.5.14	<i>Bluetooth</i> [®] RF-Transceiver.....	320
13.2.5.15	Vibration Velocity/Bearing Condition Monitoring VIB I/O.....	321
13.2.5.16	KNX/EIB/TP1 Module.....	321
13.2.5.17	AS-interface Master Module	322
13.2.6	System Modules	324
13.2.6.1	System Modules with Diagnostics.....	324
13.2.6.2	Binary Space Module	324
13.3	Process Data Architecture for EtherNet/IP	324
13.3.1	Digital Input Modules.....	326
13.3.1.1	1 Channel Digital Input Module with Diagnostics.....	326
13.3.1.2	2 Channel Digital Input Modules	326
13.3.1.3	2 Channel Digital Input Module with Diagnostics.....	327
13.3.1.4	2 Channel Digital Input Module with Diagnostics and Output Process Data.....	327
13.3.1.5	4 Channel Digital Input Modules	328
13.3.1.6	8 Channel Digital Input Modules	328
13.3.1.7	16 Channel Digital Input Modules	328
13.3.2	Digital Output Modules.....	329
13.3.2.1	1 Channel Digital Output Module with Input Process Data.....	329
13.3.2.2	2 Channel Digital Output Modules.....	330
13.3.2.3	2 Channel Digital Input Modules with Diagnostics and Input Process Data.....	330
13.3.2.4	4 Channel Digital Output Modules.....	331
13.3.2.5	4 Channel Digital Output Modules with Diagnostics and Input Process Data.....	331
13.3.2.6	8 Channel Digital Output Module	332
13.3.2.7	8 Channel Digital Output Modules with Diagnostics and Input Process Data.....	332
13.3.2.8	16 Channel Digital Output Modules.....	333
13.3.2.9	8 Channel Digital Input/Output Modules	333
13.3.3	Analog Input Modules.....	334
13.3.3.1	1 Channel Analog Input Modules.....	334
13.3.3.2	2 Channel Analog Input Modules.....	335
13.3.3.3	4 Channel Analog Input Modules.....	335
13.3.4	Analog Output Modules	336
13.3.4.1	2 Channel Analog Output Modules	336
13.3.4.2	4 Channel Analog Output Modules	336
13.3.5	Specialty Modules	337
13.3.5.1	Counter Modules	337
13.3.5.2	Pulse Width Modules.....	339
13.3.5.3	Serial Interface Modules with alternative Data Format.....	339
13.3.5.4	Serial Interface Modules with Standard Data Format	340
13.3.5.5	Data Exchange Module.....	341
13.3.5.6	SSI Transmitter Interface Modules.....	341
13.3.5.7	Incremental Encoder Interface Modules.....	342

13.3.5.8	DC-Drive Controller	344
13.3.5.9	Steppercontroller.....	345
13.3.5.10	RTC Module	346
13.3.5.11	DALI/DSI Master Module.....	347
13.3.5.12	EnOcean Radio Receiver	347
13.3.5.13	MP Bus Master Module.....	348
13.3.5.14	<i>Bluetooth</i> [®] RF-Transceiver.....	348
13.3.5.15	Vibration Velocity/Bearing Condition Monitoring VIB I/O	349
13.3.5.16	AS-interface Master Module	350
13.3.6	System Modules	351
13.3.6.1	System Modules with Diagnostics.....	351
13.3.6.2	Binary Space Module	351
14	Application Examples.....	352
14.1	Test of MODBUS protocol and fieldbus nodes	352
14.2	Visualization and Control using SCADA Software.....	352
15	Use in Hazardous Environments	355
15.1	Marking Configuration Examples.....	356
15.1.1	Marking for Europe according to CENELEC and IEC	356
15.1.2	Marking for America according to NEC 500.....	359
15.2	Installation Regulations.....	360
15.2.1	Special Conditions for Safe Operation of the ATEX and IEC Ex (acc. DEMKO 08 ATEX 142851X and IECEx PTB 07.0064).....	361
15.2.2	Special conditions for safe use (ATEX Certificate TÜV 07 ATEX 554086 X).....	362
15.2.3	Special conditions for safe use (IEC-Ex Certificate TUN 09.0001 X).....	364
15.2.4	ANSI/ISA 12.12.01	366
16	Appendix.....	368
16.1	MIB II Groups.....	368
16.1.1	System Group	368
16.1.2	Interface Group.....	369
16.1.3	IP Group	371
16.1.4	IpRoute Table Group.....	372
16.1.5	ICMP Group	373
16.1.6	TCP Group.....	374
16.1.7	UDP Group.....	375
16.1.8	SNMP Group.....	376
16.2	WAGO MIB Groups	377
16.2.1	Company Group	377
16.2.2	Product Group	377
16.2.3	Versions Group.....	378
16.2.4	Real-Time Clock Group	379
16.2.5	Ethernet Group	380
16.2.6	Actual Error Group.....	380
16.2.7	PLC Project Group	381
16.2.8	Http Group.....	382
16.2.9	Ftp Group.....	382
16.2.10	Sntp Group.....	383
16.2.11	Snmp Group.....	383

16.2.12	Snmp Trap String Group	385
16.2.13	Snmp User Trap String Group.....	386
16.2.14	Plc Connection Group	386
16.2.15	Modbus Group.....	387
16.2.16	Ethernet IP Group.....	388
16.2.17	Process Image Group.....	388
16.2.18	Plc Data Group	389
List of Figures		390
List of Tables.....		392

1 Notes about this Documentation

Note



Keep this documentation!

The operating instructions are part of the product and shall be kept for the entire lifetime of the device. They shall be transferred to each subsequent owner or user of the device. Care must also be taken to ensure that any supplement to these instructions are included, if applicable.

1.1 Validity of this Documentation

This documentation is only applicable to the device:
"Programmable Fieldbus Controller ETHERNET" 750-881 of the
WAGO-I/O-SYSTEM 750 series.

The Programmable Fieldbus Controller ETHERNET 750-881 shall only be installed and operated according to the instructions in this manual and the system description for the WAGO-I/O-SYSTEM 750.

NOTICE

Consider power layout of the WAGO-I/O-SYSTEM 750!

In addition to these operating instructions, you will also need the system description for the WAGO-I/O-SYSTEM 750, which can be downloaded at www.wago.com. There, you can obtain important information including information on electrical isolation, system power and supply specifications.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.3 Symbols

 **DANGER**

Personal Injury!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **DANGER**

Personal Injury Caused by Electric Current!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **WARNING**

Personal Injury!

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

 **CAUTION**

Personal Injury!

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE

Damage to Property Caused by Electrostatic Discharge (ESD)!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note

Important Note!

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.



Information

Additional Information:

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.4 Number Notation

Table 1: Number Notation

Number code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.5 Font Conventions

Table 2: Font Conventions

Font type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Programme\WAGO-I/O-CHECK</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualifications

All sequences implemented on Series 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

All changes to the coupler or controller should always be carried out by qualified personnel with sufficient skills in PLC programming.

2.1.3 Use of the 750 Series in Compliance with Underlying Provisions

Couplers, controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to the actuators or higher-level control systems. Using programmable controllers, the signals can also be (pre-)processed.

The components have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the components in wet and dusty environments is prohibited.

Operating 750 Series components in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section on "WAGO-I/O-SYSTEM 750" → "System Description" → "Technical Data" in the manual for the used fieldbus coupler/controller.

Appropriate housing (per 94/9/EG) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

2.1.4 Technical Condition of Specified Devices

The components to be supplied Ex Works, are equipped with hardware and software configurations, which meet the individual application requirements. WAGO Kontakttechnik GmbH & Co. KG will be exempted from any liability in case of changes in hardware or software as well as to non-compliant usage of components.

Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

2.2 Safety Advice (Precautions)

For installing and operating purposes of the relevant device to your system the following safety precautions shall be observed:



DANGER

Do not work on components while energized!

All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.

DANGER

Installation only in appropriate housings, cabinets or in electrical operation rooms!

The WAGO-I/O-SYSTEM 750 and its components are an open system. As such, install the system and its components exclusively in appropriate housings, cabinets or in electrical operation rooms. Allow access to such equipment and fixtures to authorized, qualified staff only by means of specific keys or tools.

NOTICE

Replace defective or damaged devices!

Replace defective or damaged device/module (e.g., in the event of deformed contacts), since the long-term functionality of device/module involved can no longer be ensured.

NOTICE

Protect the components against materials having seeping and insulating properties!

The components are not resistant to materials having seeping and insulating properties such as: aerosols, silicones and triglycerides (found in some hand creams). If you cannot exclude that such materials will appear in the component environment, then install the components in an enclosure being resistant to the above-mentioned materials. Clean tools and materials are imperative for handling devices/modules.

NOTICE

Cleaning only with permitted materials!

Clean soiled contacts using oil-free compressed air or with ethyl alcohol and leather cloths.

NOTICE

Do not use any contact spray!

Do not use any contact spray. The spray may impair contact area functionality in connection with contamination.

NOTICE

Do not reverse the polarity of connection lines!

Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

NOTICE



Avoid electrostatic discharge!

The devices are equipped with electronic components that you may destroy by electrostatic discharge when you touch. Pay attention while handling the devices to good grounding of the environment (persons, job and packing).

2.3 Special Use Conditions for ETHERNET Devices

If not otherwise specified, ETHERNET devices are intended for use on local networks. Please note the following when using ETHERNET devices in your system:

- Do not connect control components and control networks to an open network such as the Internet or an office network. WAGO recommends putting control components and control networks behind a firewall.
- Limit physical and electronic access to all automation components to authorized personnel only.
- Change the default passwords before first use! This will reduce the risk of unauthorized access to your system.
- Regularly change the passwords used! This will reduce the risk of unauthorized access to your system.
- If remote access to control components and control networks is required, use a Virtual Private Network (VPN).
- Regularly perform threat analyses. You can check whether the measures taken meet your security requirements.
- Use "defense-in-depth" mechanisms in your system's security configuration to restrict the access to and control of individual products and networks.

3 System Description

The WAGO-I/O-SYSTEM 750 is a modular, fieldbus-independent input/output system (I/O system). The configuration described here consists of a fieldbus coupler/controller (1) and the modular I/O modules (2) for any signal shapes that form the fieldbus node together. The end module (3) completes the node and is required for correct operation of the fieldbus node.

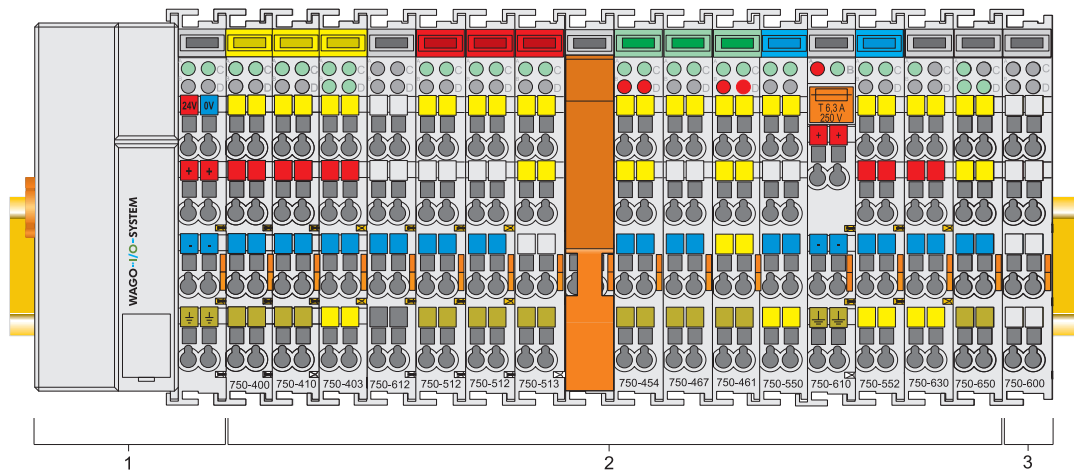


Figure 1: Fieldbus node

Couplers/controllers are available for different fieldbus systems.

The extended ECO couplers contain the fieldbus interface, electronics and a power supply terminal. The fieldbus interface forms the physical interface to the relevant fieldbus. The electronics process the data of the bus modules and make it available for the fieldbus communication. The 24 V system supply and the 24 V field supply are fed in via the integrated power supply terminal.

The coupler/controller communicates via the relevant fieldbus. The programmable fieldbus controller (PFC) enables the implementation of additional PLC functions. Programming is done with the WAGO-I/O-PRO in accordance with IEC 61131-3.

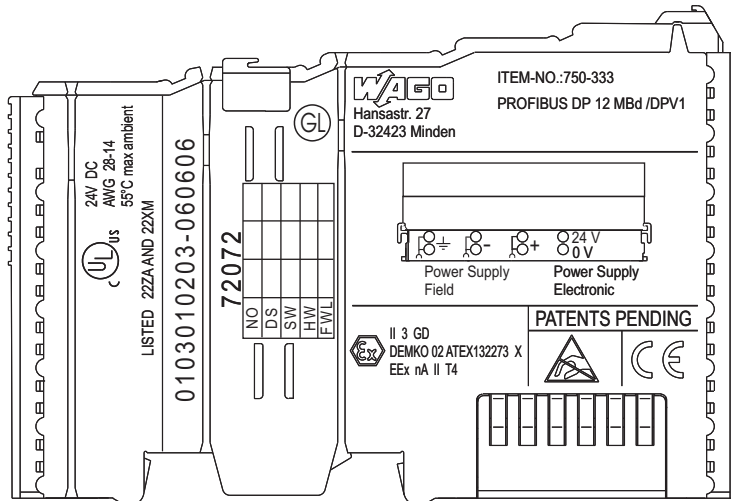
Bus modules for diverse digital and analog I/O signals as well as special functions can be connected to the fieldbus coupler/controller. The communication between the fieldbus coupler/controller and the bus modules is carried out via an internal bus.

The WAGO-I/O-SYSTEM 750 has a clear port level with LEDs for status indication, insertable mini WSB markers and pullout group marker carriers for marking.

The 1, 2 or 3 wire technology supplemented by a ground wire connection allows for direct sensor or actuator wiring.

3.1 Manufacturing Number

The serial number indicates the delivery status directly after production. This number is part of the labeling on the side of each component. In addition, the serial number is printed on the cover cap of the configuration and programming interface of the fieldbus coupler/controller, so that it can also be read when installed.



Manufacturing number

01 03 01 02 03 -B060606

Calendar	Year	Software	Hardware	Firmware	Internal
week		version	version	loader	number
				version	

Figure 2: Example of a manufacturing number

The manufacturing number consists of the production week and year, the software version (if available), the hardware version of the component, the firmware loader (if available) and further internal information for WAGO Kontakttechnik GmbH & Co. KG.

3.2 Hardware Address (MAC ID)

Each Programmable Fieldbus Controller ETHERNET has a unique and internationally unambiguous physical address, referred to as the MAC-ID (Media Access Control Identity). This is located on the rear of the controller and on a self-adhesive tear-off label on the side of the controller. The MAC ID has a set length of 6 bytes (48 bits) (hexadecimal). The first three bytes identify the manufacturer (e.g. 00:30 DE for WAGO). The second 3 bytes indicate the consecutive serial number for the hardware.

3.3 Component Update

For the case of an Update of one component, the lateral marking on each component contains a prepared matrix.

This matrix makes columns available for altogether three updates to the entry of the current update data, like production order number (NO; starting from calendar week 13/2004), update date (DS), software version (SW), hardware version (HW) and the firmware loader version (FWL, if available).

Current Version data for		1. Update	2. Update	3. Update	
Production Order Number	NO				← only starting from calendar week 13/2004
Datestamp	DS				
Software index	SW				
Hardware index	HW				
Firmware loader index	FWL				← only for coupler/controller

If the update of a component took place, the current version data are registered into the columns of the matrix.

Additionally with the update of a fieldbus coupler or controller also the cover of the configuration and programming interface of the coupler or controller is printed on with the current manufacturing and production order number.

The original manufacturing data on the housing of the component remain thereby.

3.4 Storage, Assembly and Transport

Wherever possible, the components are to be stored in their original packaging. Likewise, the original packaging provides optimal protection during transport.

When assembling or repacking the components, the contacts must not be soiled or damaged. The components must be stored and transported in appropriate containers/packaging. Thereby, the ESD information is to be regarded.

3.5 Assembly Guidelines/Standards

- DIN 60204 Electrical equipping of machines
- DIN EN 50178 Equipping of high-voltage systems with electronic components (replacement for VDE 0160)
- EN 60439 Low voltage switchgear assemblies

3.6 Power Supply

3.6.1 Isolation

Within the fieldbus node, there are three electrically isolated potentials:

- Electrically isolated fieldbus interface via transformer
- Electronics of the couplers/controllers and the bus modules (internal bus)
- All bus modules have an electrical isolation between the electronics (internal bus, logic) and the field electronics. Some digital and analog input modules have each channel electrically isolated, please see catalog.

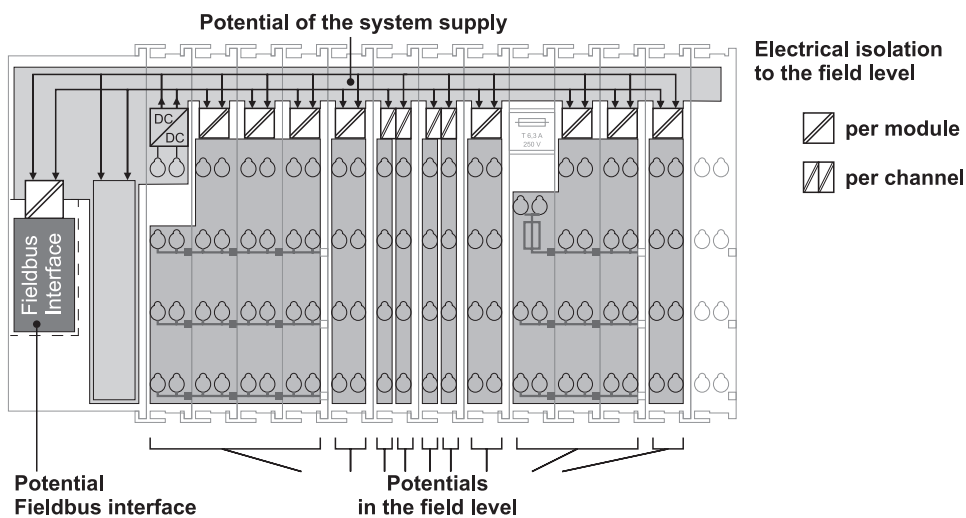


Figure 3: Isolation for Standard Couplers/Controllers and extended ECO Couplers



Note

Ensure protective conductor function is present (via ring feeding if required)!

Pay attention that the ground wire connection must be present in each group. In order that all protective conductor functions are maintained under all circumstances, it is recommended that a ground wire be connected at the beginning and the end of a potential group (please see chapter “Grounding” > “Grounding Protection”, Ring Feeding). Thus, if a bus module comes loose from a composite during servicing, therefore the protective conductor connection is still guaranteed for all connected field devices.

When you use a joint power supply unit for the 24 V system supply and the 24 V field supply, the electrical isolation between the internal bus and the field level is eliminated for the potential group.

3.6.2 System Supply

3.6.2.1 Connection

The WAGO-I/O-SYSTEM 750 requires a 24 V direct current system supply. The power supply is provided via the coupler/controller and, if necessary, in addition via the internal system supply modules (750-613). The voltage supply is reverse voltage protected.

NOTICE

Do not use an incorrect voltage/frequency!

The use of an incorrect supply voltage or frequency can cause severe damage to the component.

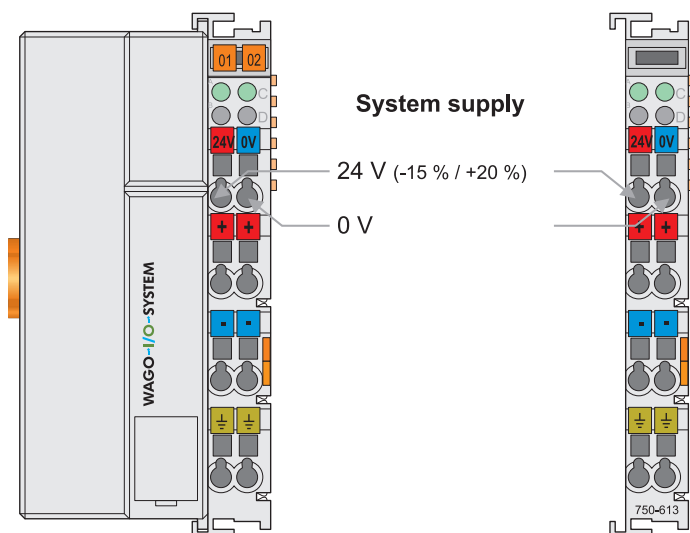


Figure 4: System supply for standard coupler/controller and extended ECO couplers

The fed DC 24 V supplies all internal system components, e.g. coupler/controller electronics, fieldbus interface and bus modules via the internal bus (5 V system voltage). The 5 V system voltage is electrically connected to the 24 V system supply.

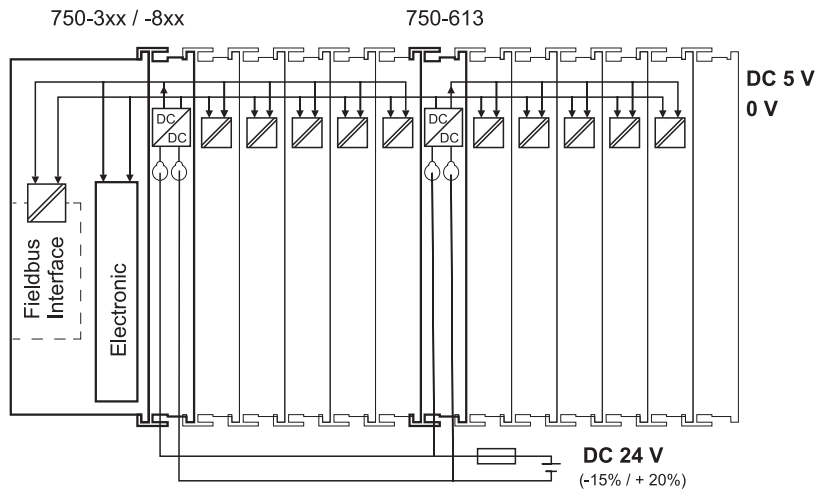


Figure 5: System voltage for standard couplers/controllers and extended ECO couplers

Note



Only reset the system simultaneously for all supply modules!

Reset the system by simultaneously switching the system supply on all supply modules (fieldbus coupler/controller and potential supply module with bus power supply 750-613) off and on again.

3.6.2.2 Dimensioning

Note



Recommendation

A stable power supply voltage cannot be taken for granted always and everywhere. Therefore, regulated power supply units should be used in order to guarantee the quality of the supply voltage.

The supply capacity of the coupler/controller or the internal system supply module (750-613) can be taken from the technical data of the components.

Table 3: Alignment

Internal current consumption*)	Current consumption via system voltage: 5 V for electronics of bus modules and coupler/controller
Total current for bus terminals*)	Available current for the bus modules. Provided by the bus power supply unit. See coupler/controller and internal system supply module (750-613)

*) See current catalog, manuals, Internet

Example:

Calculating the current consumption on an Example Coupler:

Internal current consumption	380 mA at 5 V
Residual current for bus modules	1620 mA at 5 V
Sum $I_{(5V)}$ total	2000 mA at 5V

The internal current consumption is indicated in the technical data for each bus terminal. In order to determine the total requirement, add together the values of all bus modules in the node.

Note



Observe total current of I/O modules, re-feed the potential if required!

If the sum of the internal current consumption exceeds the residual current for bus modules, then an internal system supply module (750-613) must be placed before the module where the permissible residual current was exceeded.

Example:

Calculating the total current on the Example Coupler described above:

A node with the example coupler, which is described above, consists of: 20 relay modules (750-517) and 10 digital input modules (750-405).

Internal current consumption	$20 * 90 \text{ mA} = 1800 \text{ mA}$
	$10 * 2 \text{ mA} = 20 \text{ mA}$
Sum	1820 mA

The example coupler can provide 1620 mA (see previous example) for the bus modules. This value is given in the associated data sheet. Consequently, an internal system supply module (750-613), e. g. in the middle of the node, should be added.

Note



Recommendation

You can configure with the WAGO ProServe® Software **smartDESIGNER**, the assembly of a fieldbus node. You can test the configuration via the integrated plausibility check.

The maximum input current of the 24 V system supply is 500 mA. The exact electrical consumption ($I_{(V)}$) can be determined with the following formulas:

Coupler or controller

$I_{(5\text{ V})\text{ total}}$ = Sum of all the internal current consumption of the connected bus modules + internal current consumption coupler/controller

Internal system supply module 750-613

$I_{(5\text{ V})\text{ total}}$ = Sum of all the internal current consumption of the connected bus modules at internal system supply module

$$\text{Input current } I_{(24\text{ V})} = \frac{5\text{ V}}{24\text{ V}} * \frac{I_{(5\text{ V})\text{ total}}}{\eta}$$

$$\eta = 0.87$$

(87 % Efficiency of the power supply at nominal load 24 V)

**Note****Activate all outputs when testing the current consumption!**

If the electrical consumption of the power supply point for the 24 V system supply exceeds 500 mA, then the cause may be an improperly designed node or a defect.

During the test, you must activate all outputs, in particular those of the relay modules.

3.6.3 Field Supply

3.6.3.1 Connection

Sensors and actuators can be directly connected to the relevant channel of the bus module in 1/4 conductor connection technology. The bus module supplies power to the sensors and actuators. The input and output drivers of some bus modules require the field side supply voltage.

The coupler/controller provides field side power (DC 24 V). In this case it is a passive power supply without protection equipment. Power supply modules are available for other potentials, e. g. AC 230 V.

Likewise, with the aid of the power supply modules, various potentials can be set up. The connections are linked in pairs with a power contact.

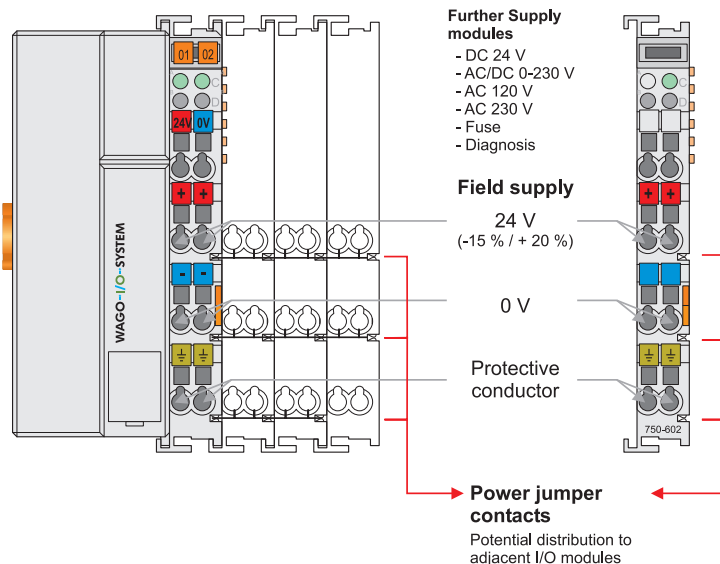


Figure 6: Field supply (sensor/actuator) for standard couplers/controllers and extended ECO couplers

The supply voltage for the field side is automatically passed to the next module via the power jumper contacts when assembling the bus modules.

The current load of the power contacts must not exceed 10 A on a continual basis. The current carrying capacity between two connection terminals is identical to the load capacity of the connection wires.

By inserting an additional power supply module, the field supply via the power contacts is disrupted. From there a new power supply occurs which may also contain a new voltage potential.

Note**Re-establish the ground connection when the connection to the power jumper contacts is disrupted!**

Some bus modules have no or very few power contacts (depending on the I/O function). Due to this, the passing through of the relevant potential is disrupted. If you require a field supply for subsequent bus modules, then you must use a power supply module.

Note the data sheets of the bus modules.

Note**Use a spacer module when setting up a node with different potentials!**

In the case of a node setup with different potentials, e.g. the alteration from DC 24 V to AC 230 V, you should use a spacer module. The optical separation of the potentials acts as a warning to heed caution in the case of wiring and maintenance works. Thus, you can prevent the results of wiring errors.

3.6.3.2 Fusing

Internal fusing of the field supply is possible for various field voltages via an appropriate power supply module.

Table 4: Power supply modules

Order No.	Field Voltage
750-601	24 V DC, Supply/Fuse
750-609	230 V AC, Supply/Fuse
750-615	120 V AC, Supply/Fuse
750-610	24 V DC, Supply/Fuse/Diagnosis
750-611	230 V AC, Supply/Fuse/Diagnosis

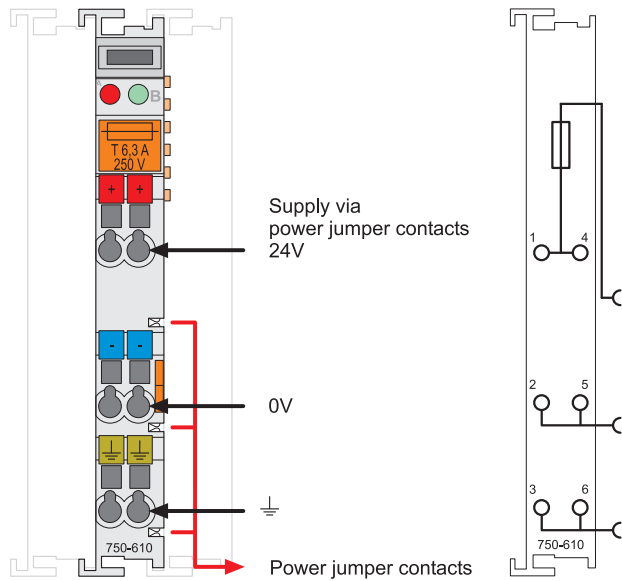


Figure 7: Supply module with fuse carrier (Example 750-610)

NOTICE

Observe the maximum power dissipation and, if required, UL requirements!
In the case of power supply modules with fuse holders, you must only use fuses with a maximum dissipation of 1.6 W (IEC 127).
For UL approved systems only use UL approved fuses.

In order to insert or change a fuse, or to switch off the voltage in succeeding bus modules, the fuse holder may be pulled out. In order to do this, use a screwdriver for example, to reach into one of the slits (one on both sides) and pull out the holder.



Figure 8: Removing the fuse carrier

Lifting the cover to the side opens the fuse carrier.

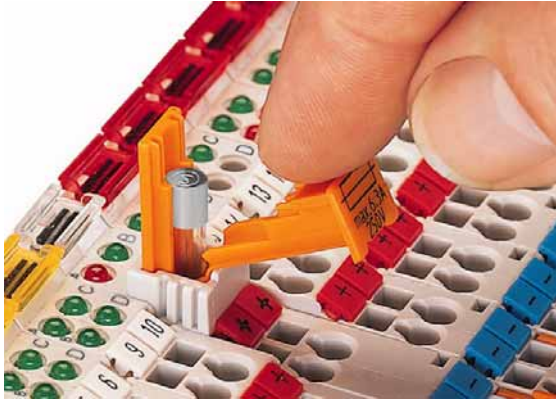


Figure 9: Opening the fuse carrier



Figure 10: Change fuse

After changing the fuse, the fuse carrier is pushed back into its original position.

Alternatively, fusing can be done externally. The fuse modules of the WAGO series 281 and 282 are suitable for this purpose.

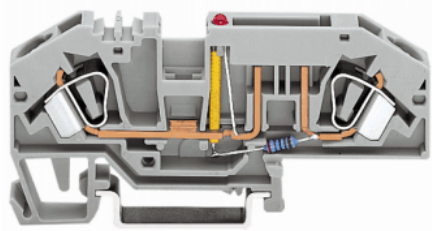


Figure 11: Fuse modules for automotive fuses, series 282

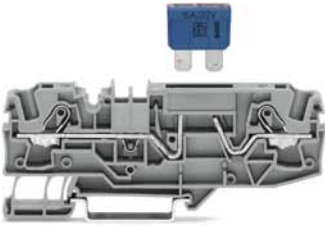


Figure 12: Fuse modules for automotive fuses, series 2006

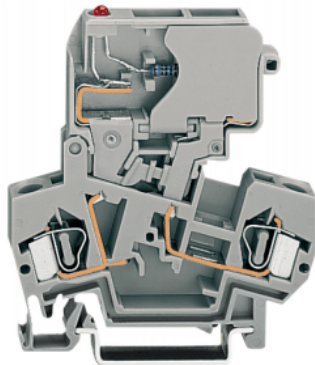


Figure 13: Fuse modules with pivotable fuse carrier, series 281

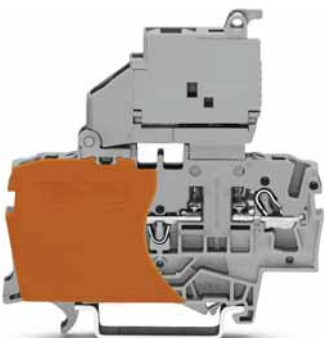


Figure 14: Fuse modules with pivotable fuse carrier, series 2002

3.6.4 Supplementary Power Supply Regulations

The WAGO-I/O-SYSTEM 750 can also be used in shipbuilding or offshore and onshore areas of work (e. g. working platforms, loading plants). This is demonstrated by complying with the standards of influential classification companies such as Germanischer Lloyd and Lloyds Register.

Filter modules for 24 volt supply are required for the certified operation of the system.

Table 5: Filter modules for 24-volt supply

Order No.	Name	Description
750-626	Supply Filter	Filter module for system supply and field supply (24 V, 0 V), i. e. for fieldbus coupler/controller and bus power supply (750-613)
750-624	Supply Filter	Filter module for the 24 V- field supply (750-602, 750-601, 750-610)

Therefore, the following power supply concept must be absolutely complied with.

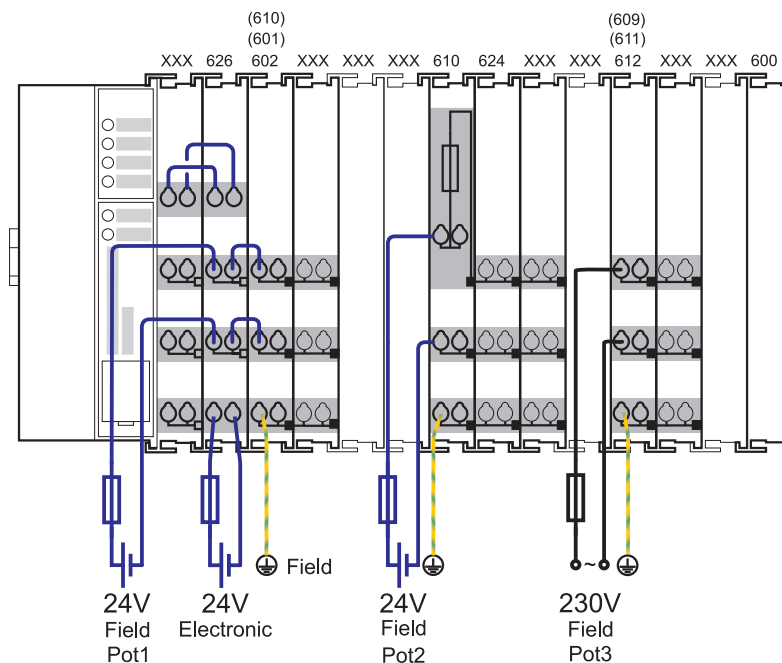


Figure 15: Power supply concept

Note



Additional supply module as ground (earth) conductor/fuse protection!
Only insert another potential power terminal 750-601/602/610 behind the filter terminal 750-626 if you need the protective earth conductor on the lower power contact or if you require a fuse protection.

3.6.5 Supply Example

Note



The system supply and the field supply shall be separated!

You should separate the system supply and the field supply in order to ensure bus operation in the event of a short-circuit on the actuator side.

Information



Additional information about the ring feeding

In order to increase the system security, a ring feeding of the earth potential is recommended. Thus, if a bus module comes loose from a composite during servicing, then the protective conductor connection is still guaranteed for all connected field devices. With the ring feeding, protective grounding is connected at the beginning and the end of a potential group.

Please refer for further information to chapter „Grounding“ > “Grounding Protection”, Ring Feeding.

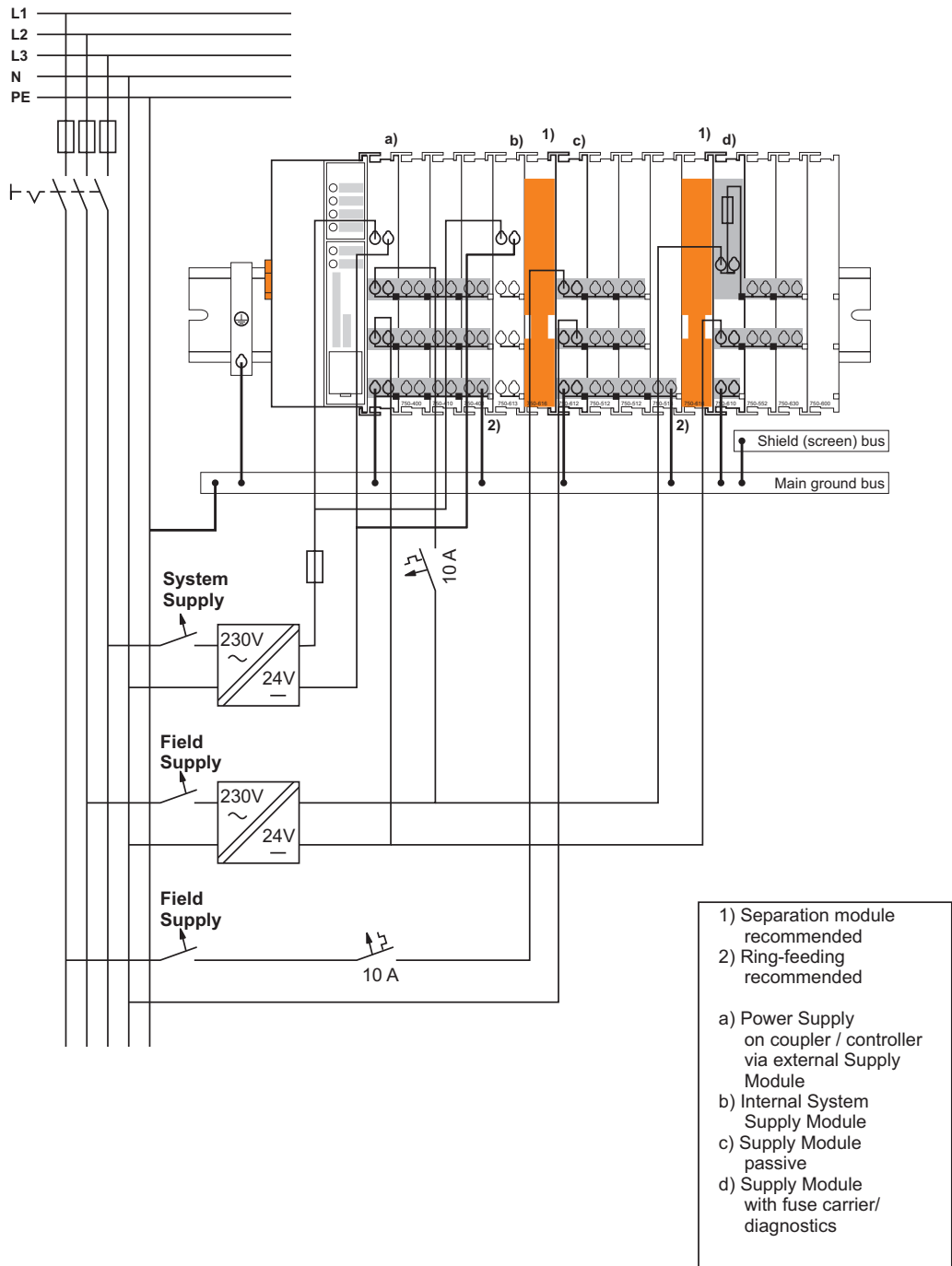


Figure 16: Supply example for standard couplers/controllers and extended ECO couplers

3.6.6 Power Supply Unit

The WAGO-I/O-SYSTEM 750 requires a 24 V direct current system supply with a maximum deviation of -15 % or +20 %.

Note



Recommendation

A stable network supply cannot be taken for granted always and everywhere. Therefore, you should use regulated power supply units in order to guarantee the quality of the supply voltage.

A buffer (200 µF per 1 A current load) should be provided for brief voltage dips.

Note



Power failure time is not acc. to IEC 61131-2!

Note that the power failure time in a node with maximal components is not 10 ms, according to the defaults of the IEC61131-2 standard.

The electrical requirement for the field supply is to be determined individually for each power supply point. Thereby all loads through the field devices and bus modules should be considered. The field supply as well influences the bus modules, as the inputs and outputs of some bus modules require the voltage of the field supply.

Note



System and field supply shall be isolated from the power supply!

You should isolate the system supply and the field supply from the power supplies in order to ensure bus operation in the event of short circuits on the actuator side.

Table 6: WAGO Power Supply Unit

WAGO Power Supply Unit	Description
787-612	Primary switched mode; DC 24 V; 2,5 A Input nominal voltage AC 230 V
787-622	Primary switched mode; DC 24 V; 5 A Input nominal voltage AC 230 V
787-632	Primary switched mode; DC 24 V; 10 A Input nominal voltage AC 230/115 V
288-809	Rail-mounted modules with universal mounting carrier AC 115 V/DC 24 V; 0,5 A
288-810	AC 230 V/DC 24 V; 0,5 A
288-812	AC 230 V/DC 24 V; 2 A
288-813	AC 115 V/DC 24 V; 2 A

3.7 Grounding

3.7.1 Grounding the DIN Rail

3.7.1.1 Framework Assembly

When setting up the framework, the carrier rail must be screwed together with the electrically conducting cabinet or housing frame. The framework or the housing must be grounded. The electrical connection is established via the screw. Thus, the carrier rail is grounded.



DANGER

Ensure sufficient grounding is provided!

You must take care to ensure the flawless electrical connection between the carrier rail and the frame or housing in order to guarantee sufficient grounding.

3.7.1.2 Insulated Assembly

Insulated assembly has been achieved when there is constructively no direct ohmic contact between the cabinet frame or machine parts and the carrier rail. Here the earth ground must be set up via an electrical conductor accordingly valid national safety regulations.



Note

Recommendation

The optimal setup is a metallic assembly plate with grounding connection with an electrical conductive link with the carrier rail.

The separate grounding of the carrier rail can be easily set up with the aid of the WAGO ground wire terminals.

Table 7: WAGO ground wire terminals

Order No.	Description
283-609	1-conductor ground (earth) terminal block make an automatic contact to the carrier rail; conductor cross section: 0.2 -16 mm ² Note: Also order the end and intermediate plate (283-320).

3.7.2 Grounding Function

The grounding function increases the resistance against disturbances from electro-magnetic interferences. Some components in the I/O system have a carrier rail contact that dissipates electro-magnetic disturbances to the carrier rail.

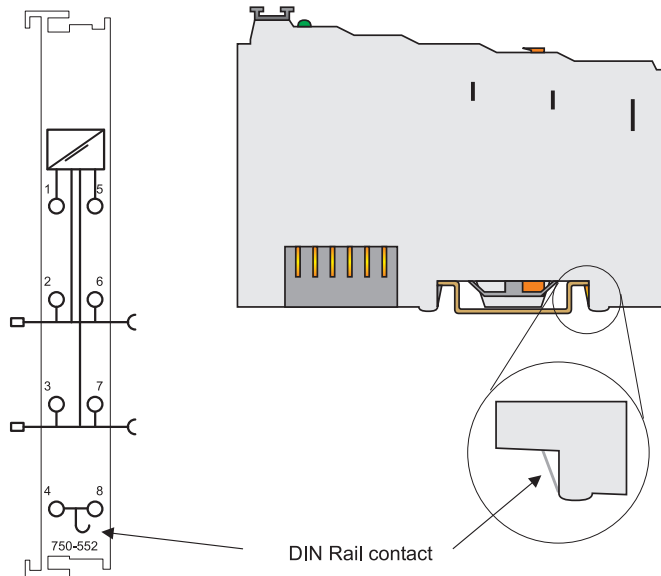


Figure 17: Carrier rail contact



DANGER

Ensure sufficient grounding is provided!

You must take care to ensure the direct electrical connection between the carrier rail contact and the carrier rail.

The carrier rail must be grounded.

For information on carrier rail properties, see chapter “Assembly onto Carrier Rail > Carrier Rail Properties“.

3.7.3 Grounding Protection

For the field side, the ground wire is connected to the lowest connection terminals of the power supply module. The ground connection is then connected to the next module via the Power Jumper Contact (PJC). If the bus module has the lower power jumper contact, then the ground wire connection of the field devices can be directly connected to the lower connection terminals of the bus module.

Note



Re-establish the ground connection when the connection to the power jumper contacts is disrupted!

Should the ground conductor connection of the power jumper contacts within the node become disrupted, e. g. due to a 4-channel bus terminal, the ground connection will need to be re-established.

The ring feeding of the grounding potential will increase the system safety. When one bus module is removed from the group, the grounding connection will remain intact.

The ring feeding method has the grounding conductor connected to the beginning and end of each potential group.

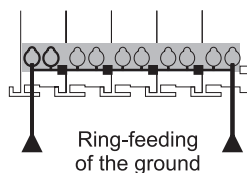


Figure 18: Ring-feeding

Note



Observe grounding protection regulations!

You must observe the regulations relating to the place of assembly as well as the national regulations for maintenance and inspection of the grounding protection.

3.8 Shielding

3.8.1 General

Use of shielded cables reduces electromagnetic interference and thus increases signal quality. Measurement errors, data transmission errors and interference due to excessive voltage can be prevented.

Note



Connect the cable shield to the ground potential!

Integrated shielding is mandatory to meet the technical specifications in regards to measuring accuracy. Connect the cable shield and ground potential at the inlet to the cabinet or housing. This allows induced interference to dissipate and to be kept away from devices in the cabinet or housing.

Note



Improve shielding performance by placing the shield over a large area!

Higher shielding performance is achieved via low-impedance connection between shield and ground. For this purpose, connect the shield over a large surface area, e.g., WAGO shield connecting system. This is especially recommended for large-scale systems where equalizing current or high impulse-type currents caused by atmospheric discharge may occur.

Note



Keep data and signal lines away from sources of interference!

Route data and signal lines separately from all high voltage cables and other sources of high electromagnetic emission (e.g., frequency converter or drives).

3.8.2 Bus cables

The shielding of the bus line is described in the respective configuration guidelines and standards of the bus system.

3.8.3 Signal lines

I/O modules for analog signals and some interface I/O modules are equipped with shield clamps.



Note

Use shielded signal lines!

Only use shielded signal lines for analog signals and I/O modules which are equipped with shield clamps. Only then can you ensure that the accuracy and interference immunity specified for the respective I/O module can be achieved even in the presence of interference acting on the signal cable.

3.8.4 WAGO Shield Connecting System

The WAGO shield connecting system consists of shield clamping saddles, busbars and various mounting carriers. These components can be used to achieve many different configurations.



Figure 19: Example of the WAGO shield connecting system



Figure 20: Application of the WAGO shield connecting system

4 Device Description

The 750-881 programmable Fieldbus Controller (PFC) combines the functionality of an ETHERNET-based Fieldbus Coupler with the functionality of a Programmable Logic Controller (PLC).

This controller can be used for applications in machine and plant construction as well as in the process industry and building technology.

The two Ethernet interfaces and the integrated switch make possible the wiring of the fieldbus in line topology. Thus additional infrastructure elements such as switches or hubs can be void. Both interfaces support Autonegotiation and Auto-MDI (X).

With the DIP switch the last byte of the IP address, as well as the assignment of the IP address (DHCP, BootP, firm setting) can be given.

In the Fieldbus Controller, all input signals from the sensors are combined. After connecting the ETHERNET TCP/IP Fieldbus Controller, the Fieldbus Controller determines which I/O modules are on the node and creates a local process image from these. Analog and specialty module data is sent via words and/or bytes; digital data is grouped bit-by-bit.

The local process image is divided into two data zones containing the data received and the data to be sent.

The data of the analog modules is mapped first into the process image. The modules are mapped in the order of their physical position after the controller.

The bits of the digital modules are combined into words and then mapped after the analog ones in the process image. If the number of digital I/Os is greater than 16 bits, the Fieldbus Controller automatically begins a new word.

According to IEC 61131-3 programming, data processing occurs in the PFC. The process results can be output directly on sensors/actuators or transmitted via fieldbus to the higher-order controller.

The fieldbus connection consists of two ports (RJ-45). An ETHERNET switch integrated in the PFC, which is operated in the store and forward mode, connects those fieldbus ports with the CPU.

Both ports support:

- 10BASE-T / 100BASE-TX
- Full / Half duplex
- Autonegotiation
- Auto-MDI(X)

WAGO-I/O-PRO creates application programs that adhere to IEC 61131-3. CoDeSys by 3S (the standard programming system) serves as the basis of WAGO-I/O-PRO, which was expanded specifically with the target files for all WAGO controllers.

The fieldbus controller has 1 MB program memory, 512 KB data memory and 32 KB retentive memory available for the IEC 61131-3 programming.

The user can access all fieldbus and I/O data.

In order to send process data via ETHERNET, the controller supports a series of network protocols.

The MODBUS/TCP(UDP) protocol and the ETHERNET/IP protocol are implemented for exchanging process data. Both of these communication protocols can be used either together or separately. For this, the write access to the I/O modules (access via PFC, MODBUS/TCP or EtherNet/IP) is specified in an xml file.

For the management and diagnosis of the system, the HTTP, SNMP and SNMP protocols are available.

For the data transfer via ETHERNET the FTP is available.

For the automatic assignment of the IP address in the network, kann alternatively DHCP or BootP can be used.

The user can program clients and servers via an internal socket-API for all transport protocols (TCP, UDP, etc.) with functional modules. Library functions are available for function expansion.

With the IEC 61131-3 library "SysLibRTC.lib," for example, a buffered real-time clock with date, time (1-second resolution), alarm functions and a timer is incorporated. This clock is supplied with auxiliary power during a power failure.

This controller is based on a 32-bit CPU with multitasking capabilities, allowing several programs to be executed in a near-simultaneous manner.

The controller has an internal server for the configuration and administration of the system.

By default, the controller's built-in HTML pages contain information on the configuration and status of the PFC, and can be read using a normal web browser. In addition, a file system is implemented that allows you to store custom HTML pages in the controller using FTP download or to store your own HTML pages or call up programs directly.

Table 8: Compatibility

Programming tool:	CoDeSys						
-Version	V2.3.9.19						
Fieldbus controller:							
750-881	✓						

Commentary:

✓	Fieldbus controller compatible with WAGO-I/O-PRO version, independent of the controller hard- or software.
---	--

4.1 View

The view below shows the three parts of the device:

- The fieldbus connection is on the left side.
- LEDs for operation status, bus communication, error messages and diagnostics, as well as the service interface are in the middle area.
- The right side shows a power supply unit for the system supply and for the field supply of the attached I/O modules via power jumper contacts. LEDs show the status of the operating voltage for the system and field supply (jumper contacts).

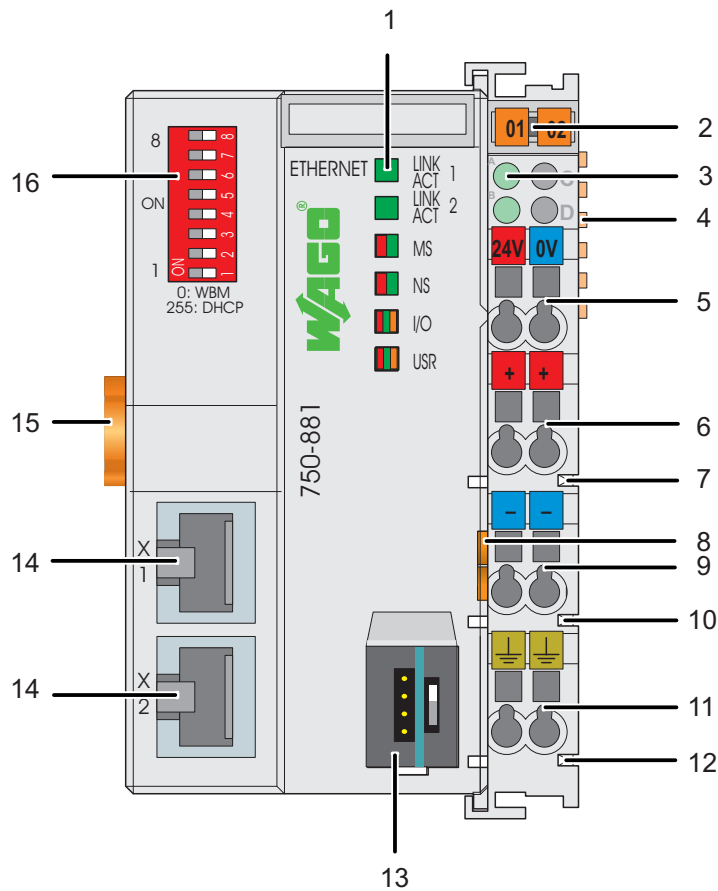


Figure 21: View ETHERNET TCP/IP Fieldbus Controller

Table 9: Legend to the View ETHERNET TCP/IP Fieldbus Controller

No.	Designation	Meaning	Details see Chapter:
1	LINK ACT 1, 2, MS, NS, I/O, USR	Status LEDs Fieldbus	"Device Description" > "Display Elements"
2	---	Group marking carrier (retractable) with additional marking possibility on two miniature WSB markers	---
3	A, B or C	Status LED's System/Field Supply	"Device Description" > "Display Elements"
4	---	Data Contacts	"Connect Devices" > "Data Contacts/Internal Bus"
5	24 V, 0 V	CAGE CLAMP® Connections System Supply	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
6	+	CAGE CLAMP® Connections Field Supply DC 24 V	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
7	---	Power Jumper Contact 24 V DC	"Connect Devices" > "Power Contacts/ Field Supply"
8	---	Unlocking Lug	"Assembly" > "Inserting and Removing Devices"
9	-	CAGE CLAMP® Connections Field Supply 0 V	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
10	---	Power Jumper Contact 0 V	"Connect Devices" > "Power Contacts/ Field Supply"
11	(Earth)	CAGE CLAMP® Connections Field Supply (Earth)	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
12	---	Power Jumper Contact (Earth)	"Connect Devices" > "Power Contacts/ Field Supply"
13	---	Service Interface (open flap)	"Device Description" > "Operating Elements"
14	X1, X2	Fieldbus connection 2 x RJ-45 as 2-Port ETHERNET Switch	"Device Description" > "Connectors"
15	---	Locking Disc	"Assembly" > "Inserting and Removing Devices"
16	---	Address Selection Switch	"Device Description" > "Operating Elements"

4.2 Connectors

4.2.1 Device Supply

The device is powered via terminal blocks with CAGE CLAMP® connections. The device supply generates the necessary voltage to power the electronics of the device and the internal electronics of the connected I/O modules.

The fieldbus interface is galvanically separated from the electrical potential of the device.

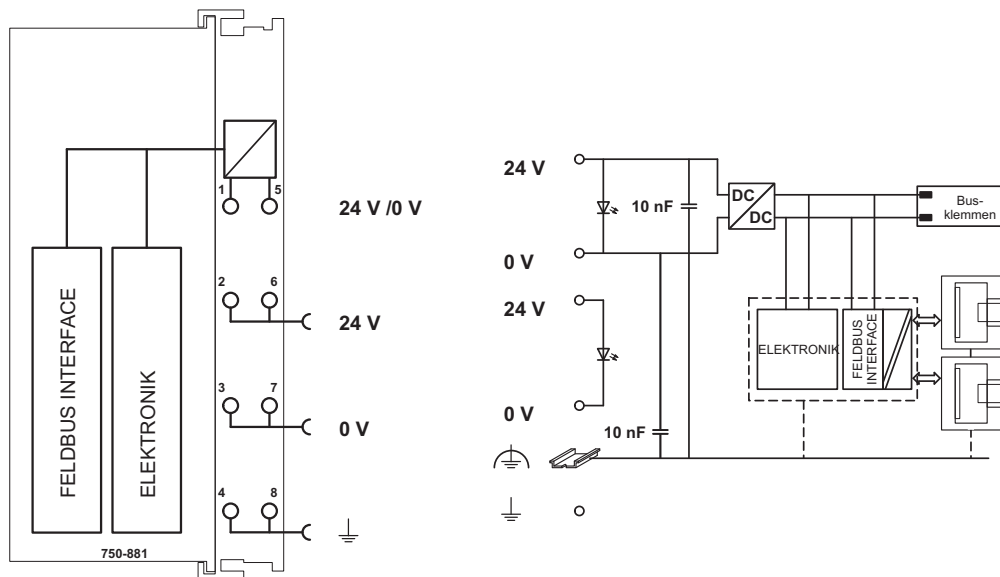


Figure 22: Device Supply

4.2.2 Fieldbus Connection

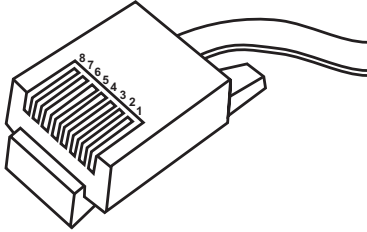
The connection to the fieldbus is made via two RJ-45 plugs (also called "Western plugs"), which are connected to the fieldbus controller via an integrated switch.

The integrated switch works in store-and-forward operation and for each port, supports the transmission speeds 10/100 Mbit as well as the transmission modes full and half-duplex.

The wiring of these plugs corresponds to the specifications for 100BaseTX, which prescribes a category 5 twisted pair cable as the connecting cable. Cable types S-UTP (Screened Unshielded Twisted Pair) and STP (Shielded Twisted Pair) with a maximum segment length of 100 m (approximately 328.08 feet) can be used.

The RJ-45 socket is physically lower, allowing the coupler to fit in an 80 mm high enclosure once connected.

Table 10: RJ-45 Connector and RJ-45 Connector Configuration

View	Contact	Signal	
		1	TD +
 <p>Figure 23: RJ-45-Connector</p>	2	TD -	Transmit -
	3	RD +	Receive +
	4		free
	5		free
	6	RD -	Receive -
	7		free
	8		free

NOTICE

Not for use in telecommunication circuits!

Only use devices equipped with ETHERNET or RJ-45 connectors in LANs.
Never connect these devices with telecommunication networks.

4.3 Display Elements

The operating condition of the controller or the node is displayed with the help of illuminated indicators in the form of light-emitting diodes (LEDs). The LED information is routed to the top of the case by light fibres. In some cases, these are multi-colored (red, green or red/green (=orange)).

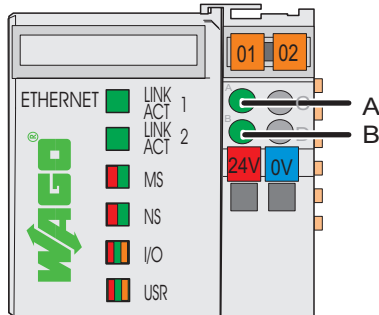


Figure 24: Display Elements

For the diagnostics of the different ranges fieldbus, node and supply voltage, the LED's can be divided into three groups:

Table 11: Display Elements Fieldbus Status

LED	Color	Meaning
LINK ACT 1	green	indicates a connection to the physical network at port 1
LINK ACT 2	green	indicates a connection to the physical network at port 2
MS	red/green	indicates the status of the node
NS	red/green	indicates the network status

Table 12: Display Elements Node Status

LED	Color	Meaning
I/O	red/green/ orange	indicates the operation of the node and signals via a blink code faults encountered
USR	red/green/ orange	indicates information to the Internal bus faults, controlled from the user programm according to the visualization programming.

Table 13: Display Elements Supply Voltage

LED	Color	Meaning
A	green	indicates the status of the operating voltage – system
B	green	indicates the status of the operating voltage – power jumper contacts

Information



More information about the LED Signaling

Read the detailed description for the evaluation of the displayed LED-Signals in the chapter "Diagnostics" > "LED Signaling".

4.4 Operating Elements

4.4.1 Service Interface

The Service Interface is to find behind the flap.

It is used for the communication with WAGO-I/O-CHECK, WAGO-I/O-PRO and for downloading firmware.

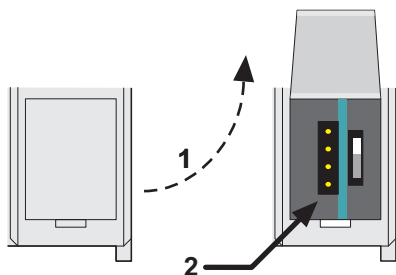


Fig. 25: Service interface for programming and configuration (closed and open door)

Table 14: Service port

Number	Description
1	Open the damper
2	Configuration and Programming Interface

NOTICE

Device must be de-energized!

To prevent damage to the device, unplug and plug in the communication cable only when the device is de-energized!

The 750-920 or 750-923 Communication Cable is connected to the 4-pole header.

4.4.2 Mode Selector Switch

The mode selector switch is located behind the cover flap.

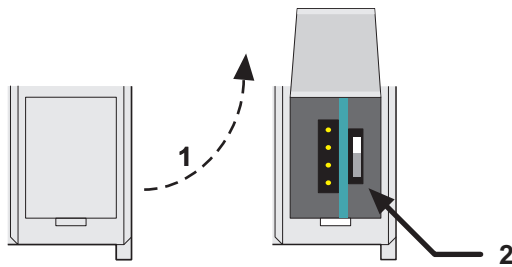


Figure 26: Mode selector switch (closed and open damper of the service port)

Table 15: Mode selector switch

Number	Description
1	Open the damper
2	Operating mode switch

The operating mode switch determines the loading, starting and stopping of the PLC-application by the controller. This multifunction sliding switch features 3 slide lock positions and a push-button function.

The sliding switch is designed for a number of operations in compliance with EN61131T2.

NOTICE

Property damages due to set outputs!

Please note that set outputs remain set, when you switch the operating switch from "RUN" to "STOP" during the current operation. Since the program is no longer processed, software-related switch offs, i.e. by initiators, are ineffective. Therefore, program or define all outputs, so that these switch to a safe mode at a program stop.

Note



Defining the outputs for a program stop!

In order to switch the outputs to a safe mode at the program stop, define the status of the outputs at "STOP".

- For this, open in the web-based Management System (WBM) a website via the "PLC" link, on which you can define the function ***Process image - Set outputs to zero, if user program is stopped.***
- Now activate this function by placing a check mark in the control box, then all outputs are set to zero, if this function is not activated, the outputs remain at the last current value.



Note

Mode selector switch position is negligible in software start/stop!

The position of the mode selector switch is not important when starting or stopping the PFC application from *WAGO-I/O-PRO*.

One of the following functions is active, depending in which of the three static positions "top", "center" or "bottom" the switch is located at a power on or in a hardware or software reset:

Table 1: Mode selector switch positions, static positions at Power On / reset

Positions of the mode selector switch	Function
Up position	"RUN" - activate program processing, Boot- project (if available) is started.
Center position	"STOP" - stop program processing, PFC- application is stopped.
Down position	After a PowerOn reset, the controller is in Bootstrap mode.

The controller performs the following functions, if a position change of the switch is performed during the current operation:

Table 2: Mode selector switch positions, dynamic positions during the current operation

Position change of the mode selector switch	Function
From the top to the center position	"STOP" - stop program processing, PFC- application is stopped.
From the center to the top position	"RUN" - activate program processing, Boot project (if available) is started.
From the center to the bottom position	No reaction. After Power On/Reset the Bootstrap loader is started on the service interface.
From the bottom to the center position	No reaction.
Press down (e.g., using a screwdriver)	Hardware reset All outputs are reset; variables are set to 0, FALSE or to an initial value. Retain variables or markers are not changed. A hardware reset can be performed either at STOP or at RUN at any position of the mode selector switch. Restart the fieldbus controller.

The operating mode is changed internally at the end of a PFC cycle.

4.4.3 Address Selection Switch

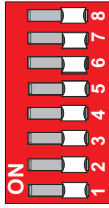


Figure 27: Address Selection Switch

The 8-pole DIP switch is used to set the IP address and to select the protocol for setting the IP address.

Table 16: Meaning of DIP switch positions

Address	Meaning
0	The IP parameter is configured via the web-based management. BootP, DHCP and application of the values from the EEPROM are available to the user. In the default status, configuration via BootP is activated.
1-254	The configuration of the IP address consists of the network address (configurable and 192.168.1 by default) and the value set for the DIP switch.
255	The DHCP protocol is used to configure the IP parameters.

4.5 Technical Data

4.5.1 Device Data

Table 17: Technical data – Device data

Width	62 mm
Height (from upper-edge of DIN 35)	65 mm
Length	100 mm
Weight	approx. 160 g
Degree of protection	IP 20

4.5.2 System Data

Table 18: Technical data – System data

Number of controllers	Limited by ETHERNET specification; max. 20 x 750-881 series connected
Transmission medium	Twisted Pair S/UTP, STP 100 Ω Cat 5
Bus coupler connection	RJ-45
Max. length of fieldbus segment	100 m behind hub and 750-881
Max. length of network	2000 m
Baud rate	10/100 Mbit/s
Protocols	MODBUS/TCP (UDP), ETHERNET/IP, HTTP, BootP, DHCP, DNS, SNTP, FTP, SNMP
Programming	WAGO-I/O-PRO
IEC-61131-3	AWL, KOP, FUP (CFC), ST, AS
Max. number of socket links	3 HTTP, 15 MODBUS/TCP, 10 FTP, 2 SNMP, 5 for IEC-61131-3 program, 2 for WAGO-I/O-PRO, 128 for Ethernet/IP
Powerfail RTC Buffer	at least 6 days
Number of I/O modules - with bus extension	64 250
Configuration	via PC
Program memory	1 Mbyte
Data memory	512 kByte
Non-volatile memory (retain)	32 kByte (16 kByte retain, 16 kByte flag)

4.5.3 Supply

Table 19: Technical data – Supply

Voltage supply	DC 24 V (-25 % ... +30 %)
Input current _{max.}	500 mA at 24 V
Efficiency of the power supply	90 %
Internal current consumption	450 mA at 5 V
Total current for I/O modules	1700 mA at 5 V
Isolation	500 V system/supply
Voltage via power jumper contacts	DC 24 V (-25 % ... +30 %)
Current via power jumper contacts _{max.}	DC 10 A

4.5.4 Fieldbus MODBUS/TCP

Table 20: Technical data – Fieldbus MODBUS/TCP

Input process image _{max.}	2040 Byte
Output process image _{max.}	2040 Byte
Input variables _{max.}	512 Byte
Output variables _{max.}	512 Byte

4.5.5 Accessories

Table 21: Technical data – Accessories

Miniature WSB Quick marking system
WAGO-I/O-PRO

4.5.6 Wire Connection

Table 22: Technical Data Wire Connection

Wire connection	CAGE CLAMP®
Cross section	0.08 mm ² ... 2.5 mm ² , AWG 28-14
Stripped lengths	8 ... 9 mm / 0.33 in
Power jumper contacts	blade/spring contact, self-cleaning
Voltage drop at I _{max.}	< 1 V/64 modules
Data contacts	slide contact, hard gold plated 1.5 µm, self-cleaning

4.5.7 Climatic environmental conditions

Table 23: Technical Data - Climatic environmental conditions

Operating temperature range	0 °C ... 55 °C
Storage temperature range	-25 °C ... +85 °C
Relative humidity without condensation	max. 95 %
Resistance to harmful substances	Acc. to IEC 60068-2-42 and IEC 60068-2-43
Maximum pollutant concentration at relative humidity < 75%	SO ₂ ≤ 25 ppm H ₂ S ≤ 10 ppm
Special conditions	Ensure that additional measures for components are taken, which are used in an environment involving: – dust, caustic vapors or gases – ionizing radiation

NOTICE

Reduced buffer time at high storage temperature!

Ensure that the storage of devices with a real time clock at high temperatures leads to a reduced buffer time for the real time clock.

4.5.8 Mechanical strength

Table 24: Technical data – Mechanical strength

Vibration resistance	acc. to IEC 60068-2-6 Comment to the vibration resistance: a) Type of oscillation: sweep with a rate of change of 1 octave per minute 10 Hz ≤ f < 57 Hz, const. Amplitude 0,075 mm 57 Hz ≤ f < 150 Hz, const. Acceleration 1 g b) Period of oscillation: 10 sweep per axis in each of the 3 vertical axes
Shock resistance	acc. to IEC 60068-2-27 Comment to the shock resistance: a) Type of impulse: half sinusoidal b) Intensity of impulse: 15 g peak value, 11 ms maintenance time c) Route of impulse: 3 impulses in each pos. And neg. direction of the 3 vertical axes of the test object, this means 18 impulses in all
Free fall	acc. IEC 60068-2-32 ≤ 1m (module in original packing)

4.6 Approvals



Information

More Information about Approvals



Detailed references to the approvals are listed in the document "Overview Approvals **WAGO-I/O-SYSTEM 750**", which you can find on the DVD "AUTOMATION Tools and Docs" (order no. 0888-0412) or via the internet under: www.wago.com → Documentation → WAGO-I/O-SYSTEM 750 → System Description.

The following approvals have been granted to 750-881 fieldbus coupler/controller:

 Conformity Marking

 cUL_{US} (UL508)

The following Ex approvals have been granted to 750-881 fieldbus coupler/controller:


 TÜV 07 ATEX 554086 X
 I M2 Ex db I Mb
 II 3 G Ex nAc IIC T4 Gc
 II 3 D Ex tc IIIC T135°C Dc

Permissible operation temperature: $0\text{ °C} \leq T_A \leq +60\text{ °C}$

 TÜV TUN 09.0001X
 Ex db I Mb
 Ex nAc IIC T4 Gc
 Ex tc IIIC T135°C Dc

Permissible operation temperature: $0\text{ °C} \leq T_A \leq +60\text{ °C}$

The following ship approvals are pending for 750-881 fieldbus coupler/controller:

 GL (Germanischer Lloyd) Cat. A, B, C, D (EMC 1)



Information

For more information about the ship approvals:

Note the "Supplementary Power Supply Regulations" chapter for the ship approvals.

4.7 Standards and Guidelines

750-881 meets the following requirements on emission and immunity of interference:

EMC CE-Immunity to interference	acc. to EN 61000-6-2: 2005
EMC CE-Emission of interference	acc. to EN 61000-6-3: 2007
EMC marine applications-Immunity to interference	acc. to Germanischer Lloyd (2003)
EMC marine applications-Emission of interference	acc. to Germanischer Lloyd (2003)

5 Assembly

5.1 Installation Position

Along with horizontal and vertical installation, all other installation positions are allowed.

Note



Use an end stop in the case of vertical assembly!

In the case of vertical assembly, an end stop has to be mounted as an additional safeguard against slipping.

WAGO order no. 249-116 End stop for DIN 35 rail, 6 mm wide

WAGO order no. 249-117 End stop for DIN 35 rail, 10 mm wide

5.2 Total Extension

The length of the module assembly (including one end module of 12mm width) that can be connected to the 750-881 is 780 mm. When assembled, the I/O modules have a maximum length of 768 mm.

Examples:

- 64 I/O modules of 12 mm width can be connected to one coupler/controller.
- 32 I/O modules of 24 mm width can be connected to one coupler/controller.

Exception:

The number of connected I/O modules also depends on which type of coupler/controller is used. For example, the maximum number of I/O modules that can be connected to a PROFIBUS coupler/controller is 63 without end module.

NOTICE

Observe maximum total length of a node!

The maximum total length of a node without a coupler/controller must not exceed 780 mm. Furthermore, you must observe restrictions made on certain types of couplers/controllers.

Note



Increase total length using a WAGO internal data bus extension module!

Using an internal data bus extension module from WAGO, you can increase the total length of the fieldbus node. In this type of configuration, you must connect a 750-627 Bus Extension End Module to the last module of the node.

You then connect the 750-627 module to the 750-628 Coupler Module of the next I/O module assembly via RJ-45 cable.

You can connect up to 10 internal data bus extension coupler modules 750-628 to an internal data bus extension end module 750-627. In this manner, you can logically connect up to 10 module assemblies to a 750-881, dividing a fieldbus node into 11 assemblies maximum.

The maximum cable length between two assemblies is 5 meters. For additional information, refer to the "750-627/-628 Modules" manual. The total cable length for a fieldbus node is 70 meters.

5.3 Assembly onto Carrier Rail

5.3.1 Carrier Rail Properties

All system components can be snapped directly onto a carrier rail in accordance with the European standard EN 50022 (DIN 35).

NOTICE

Do not use any third-party carrier rails without approval by WAGO!

WAGO Kontakttechnik GmbH & Co. KG supplies standardized carrier rails that are optimal for use with the I/O system. If other carrier rails are used, then a technical inspection and approval of the rail by WAGO Kontakttechnik GmbH & Co. KG should take place.

Carrier rails have different mechanical and electrical properties. For the optimal system setup on a carrier rail, certain guidelines must be observed:

- The material must be non-corrosive.
- Most components have a contact to the carrier rail to ground electro-magnetic disturbances. In order to avoid corrosion, this tin-plated carrier rail contact must not form a galvanic cell with the material of the carrier rail which generates a differential voltage above 0.5 V (saline solution of 0.3 % at 20°C).
- The carrier rail must optimally support the EMC measures integrated into the system and the shielding of the bus module connections.
- A sufficiently stable carrier rail should be selected and, if necessary, several mounting points (every 20 cm) should be used in order to prevent bending and twisting (torsion).
- The geometry of the carrier rail must not be altered in order to secure the safe hold of the components. In particular, when shortening or mounting the carrier rail, it must not be crushed or bent.
- The base of the I/O components extends into the profile of the carrier rail. For carrier rails with a height of 7.5 mm, mounting points are to be riveted under the node in the carrier rail (slotted head captive screws or blind rivets).
- The medal springs on the bottom of the housing must have low-impedance contact with the DIN rail (wide contact surface is possible).

5.3.2 WAGO DIN Rail

WAGO carrier rails meet the electrical and mechanical requirements shown in the table below.

Table 25: WAGO DIN Rail

Order number	Description
210-113 /-112	35 x 7,5; 1 mm; steel yellow chromated; slotted/unslotted
210-114 /-197	35 x 15; 1,5 mm; steel yellow chromated; slotted/unslotted
210-118	35 x 15; 2,3 mm; steel yellow chromated; unslotted
210-198	35 x 15; 2,3 mm; copper; unslotted
210-196	35 x 7,5; 1 mm; aluminum; unslotted

5.4 Spacing

The spacing between adjacent components, cable conduits, casing and frame sides must be maintained for the complete fieldbus node.

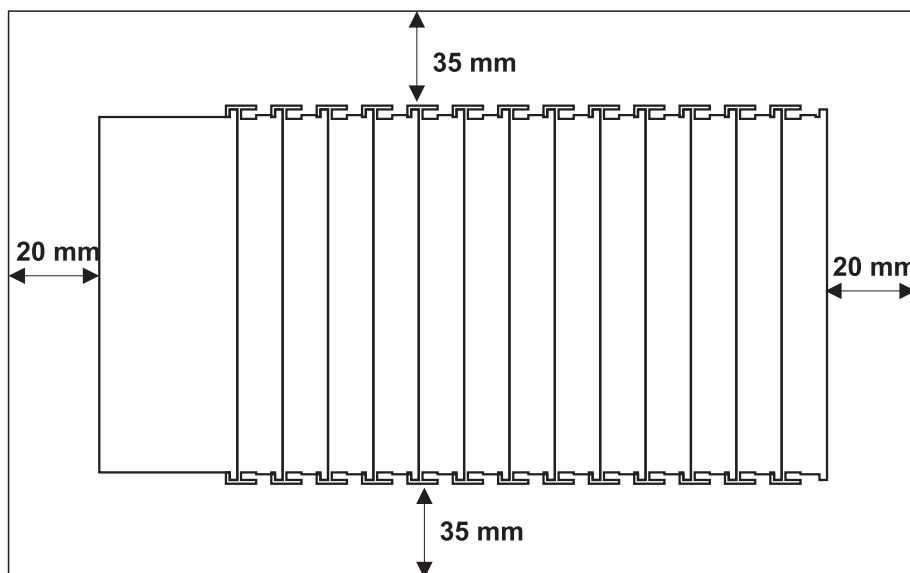


Figure 28: Spacing

The spacing creates room for heat transfer, installation or wiring. The spacing to cable conduits also prevents conducted electromagnetic interferences from influencing the operation.

5.5 Assembly Sequence

All system components can be snapped directly on a carrier rail in accordance with the European standard EN 50022 (DIN 35).

The reliable positioning and connection is made using a tongue and groove system. Due to the automatic locking, the individual components are securely seated on the rail after installation.

Starting with the coupler/controller, the bus modules are assembled adjacent to each other according to the project design. Errors in the design of the node in terms of the potential groups (connection via the power contacts) are recognized, as the bus modules with power contacts (male contacts) cannot be linked to bus modules with fewer power contacts.

CAUTION

Risk of injury due to sharp-edged male contacts!

The male contacts are sharp-edged. Handle the module carefully to prevent injury.

NOTICE

Connect the I/O modules in the required order!

Never plug bus modules from the direction of the end terminal. A ground wire power contact, which is inserted into a terminal without contacts, e.g. a 4-channel digital input module, has a decreased air and creepage distance to the neighboring contact in the example DI4.

NOTICE

Assemble the I/O modules in rows only if the grooves are open!

Please take into consideration that some bus modules have no or only a few power jumper contacts. The design of some modules does not allow them to be physically assembled in rows, as the grooves for the male contacts are closed at the top.

Note



Don't forget the bus end module!

Always plug a bus end module 750-600 onto the end of the fieldbus node! You must always use a bus end module at all fieldbus nodes with the WAGO I/O System 750 fieldbus couplers/controllers to guarantee proper data transfer.

5.6 Inserting and Removing Devices

DANGER

Use caution when interrupting the PE!

Make sure that people or equipment are not placed at risk when removing an I/O module and the associated PE interruption. To prevent interruptions, provide ring feeding of the ground conductor, see section "Grounding/Ground Conductor" in manual "System Description WAGO-I/O-SYSTEM 750".

NOTICE

Perform work on devices only if the system is de-energized!

Working on devices when the system is energized can damage the devices. Therefore, turn off the power supply before working on the devices.

5.6.1 Inserting the Fieldbus Coupler/Controller

1. When replacing the fieldbus coupler/controller for an already available fieldbus coupler/controller, position the new fieldbus coupler/controller so that the tongue and groove joints to the subsequent I/O module are engaged.
2. Snap the fieldbus coupler/controller onto the carrier rail.
3. Use a screwdriver blade to turn the locking disc until the nose of the locking disc engages behind the carrier rail (see the following figure). This prevents the fieldbus coupler/controller from canting on the carrier rail.

With the fieldbus coupler/controller snapped in place, the electrical connections for the data contacts and power contacts (if any) to the possible subsequent I/O module are established.

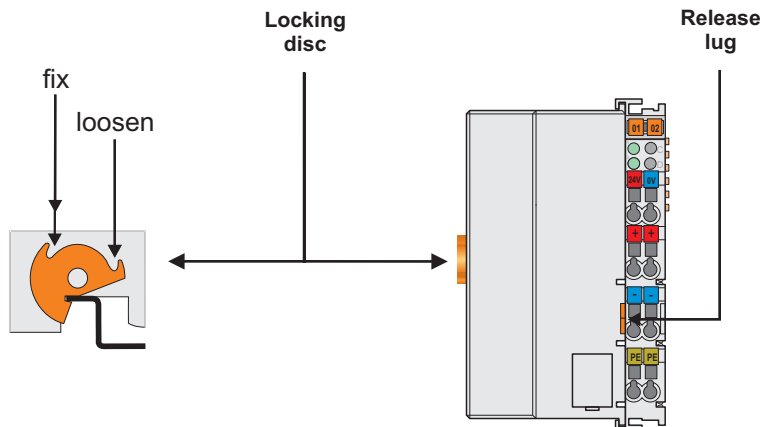


Figure 29: Unlocking lug of extended ECO coupler

5.6.2 Removing the Fieldbus Coupler/Controller

1. Use a screwdriver blade to turn the locking disc until the nose of the locking disc no longer engages behind the carrier rail.
2. Remove the fieldbus coupler/controller from the assembly by pulling the release tab.

Electrical connections for data or power contacts to adjacent I/O modules are disconnected when removing the fieldbus coupler/controller.

5.6.3 Inserting I/O Module

1. Position the I/O module so that the tongue and groove joints to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are engaged.

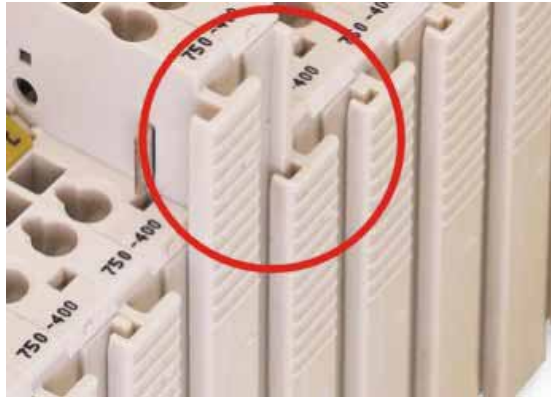


Figure 30: Insert I/O module

2. Press the I/O module into the assembly until the I/O module snaps into the carrier rail.

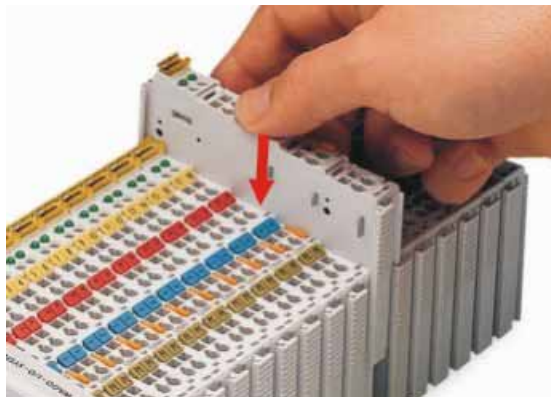


Figure 31: Snap the I/O module into place

With the I/O module snapped in place, the electrical connections for the data contacts and power contacts (if any) to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are established.

5.6.4 Removing the I/O Module

1. Remove the I/O module from the assembly by pulling the release tab.

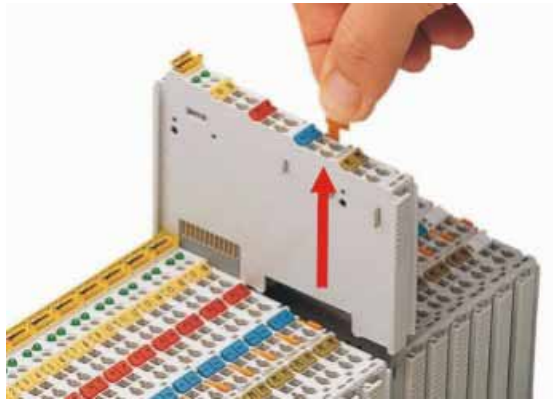


Figure 32: Removing the I/O module

Electrical connections for data or power contacts are disconnected when removing the I/O module.

6 Connect Devices

6.1 Data Contacts/Internal Bus

Communication between the coupler/controller and the bus modules as well as the system supply of the bus modules is carried out via the internal bus. It is comprised of 6 data contacts, which are available as self-cleaning gold spring contacts.

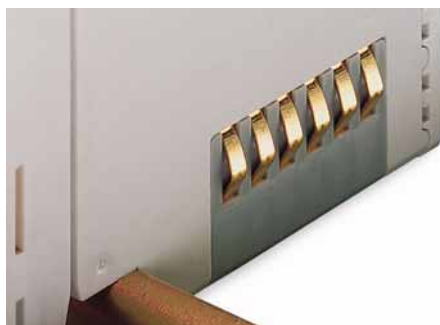


Figure 33: Data contacts

NOTICE

Do not place the I/O modules on the gold spring contacts!

Do not place the I/O modules on the gold spring contacts in order to avoid soiling or scratching!

NOTICE



Ensure that the environment is well grounded!

The modules are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the modules, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, e.g. data contacts.

6.2 Power Contacts/Field Supply

⚠ CAUTION

Risk of injury due to sharp-edged male contacts!

The male contacts are sharp-edged. Handle the module carefully to prevent injury.

Self-cleaning power jumper contacts used to supply the field side are located on the right side of both couplers/controllers and some of the I/O modules. These contacts come as touch-proof spring contacts. As fitting counterparts the I/O modules have male contacts on the left side.

Power jumper contacts

Blade	0	0	3	2
Spring	0	3	3	2

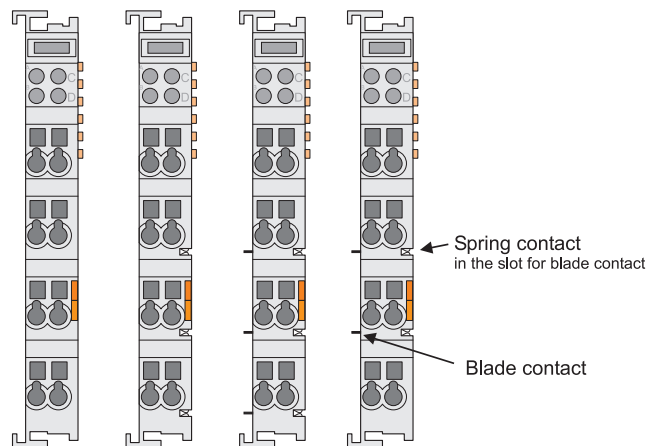


Figure 34: Example for the arrangement of power contacts

Note



Field bus node configuration and test via smartDESIGNER

With the WAGO ProServe® Software smartDESIGNER, you can configure the structure of a field bus node. You can test the configuration via the integrated accuracy check.

6.3 Connecting a conductor to the CAGE CLAMP®

The WAGO CAGE CLAMP® connection is appropriate for solid, stranded and finely stranded conductors.

Note



Only connect one conductor to each CAGE CLAMP® connection!

Only one conductor may be connected to each CAGE CLAMP® connection.

Do not connect more than one conductor at one single connection!

If more than one conductor must be routed to one connection, these must be connected in an up-circuit wiring assembly, for example using WAGO feed-through terminals.

Exception:

If it is unavoidable to jointly connect 2 conductors, then you must use a ferrule to join the wires together. The following ferrules can be used:

Length	8 mm
Nominal cross section _{max.}	1 mm ² for 2 conductors with 0.5 mm ² each
WAGO Product	216-103 or products with comparable properties.

1. To open the CAGE CLAMP® insert the actuating tool into the opening above the connection.
2. Insert the conductor into the corresponding connection opening.
3. To close the CAGE CLAMP® simply remove the tool - the conductor is then clamped firmly in place.

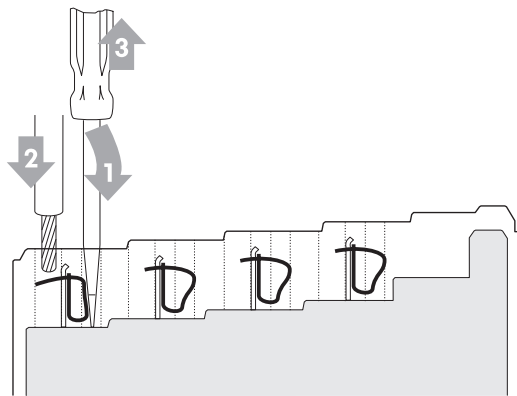


Figure 35: Connecting a conductor to a CAGE CLAMP®

7 Function Description

7.1 Operating System

7.1.1 Run-up



Note

The mode selector switch may not be located in the lower position!
The mode selector switch may not be set at the bottom position during run-up!

The controller begins running up after switching on the power supply or after a reset. The internal PFC program is then transferred to the RAM.

During the initialization phase, the fieldbus controller detects the I/O modules and the current configuration and sets the variables to 0 or FALSE, or to an initial value specified by the PFC program. The flags retain their status. During this phase the I/O LED will flash red.

When run-up is successful, the I/O LED then stays lit continuously in green.

7.1.2 PFC Cycle

After error-free run-up, the PFC cycle starts with the mode selector switch at the top position, or on a Start command from WAGO-I/O-PRO. The input and output data for the field bus, I/O modules and the timer values are read. The PFC program contained in the RAM is then processed, after which the output data for the field bus and I/O modules is written to the process image. At the end of the PFC cycle, the operating system functions are executed for diagnostics and communication (among other things) and the timer values are updated. The new cycle begins by reading in of the input and output data and the timer values.

The operating mode is changed ("STOP"/"RUN") at the end of a PFC cycle.

The cycle time is the time from the beginning of the PFC program up to the next beginning of the cycle. If a loop is programmed within the PFC program, the PFC runtime and the PFC cycle time will be extended accordingly.

The inputs, outputs and timer values are not updated while the PFC program is being processed. Updating is performed only as defined at the end of the PFC program. As a result, it is not possible to wait on an event from the process or a set period to expire while a loop is in progress.

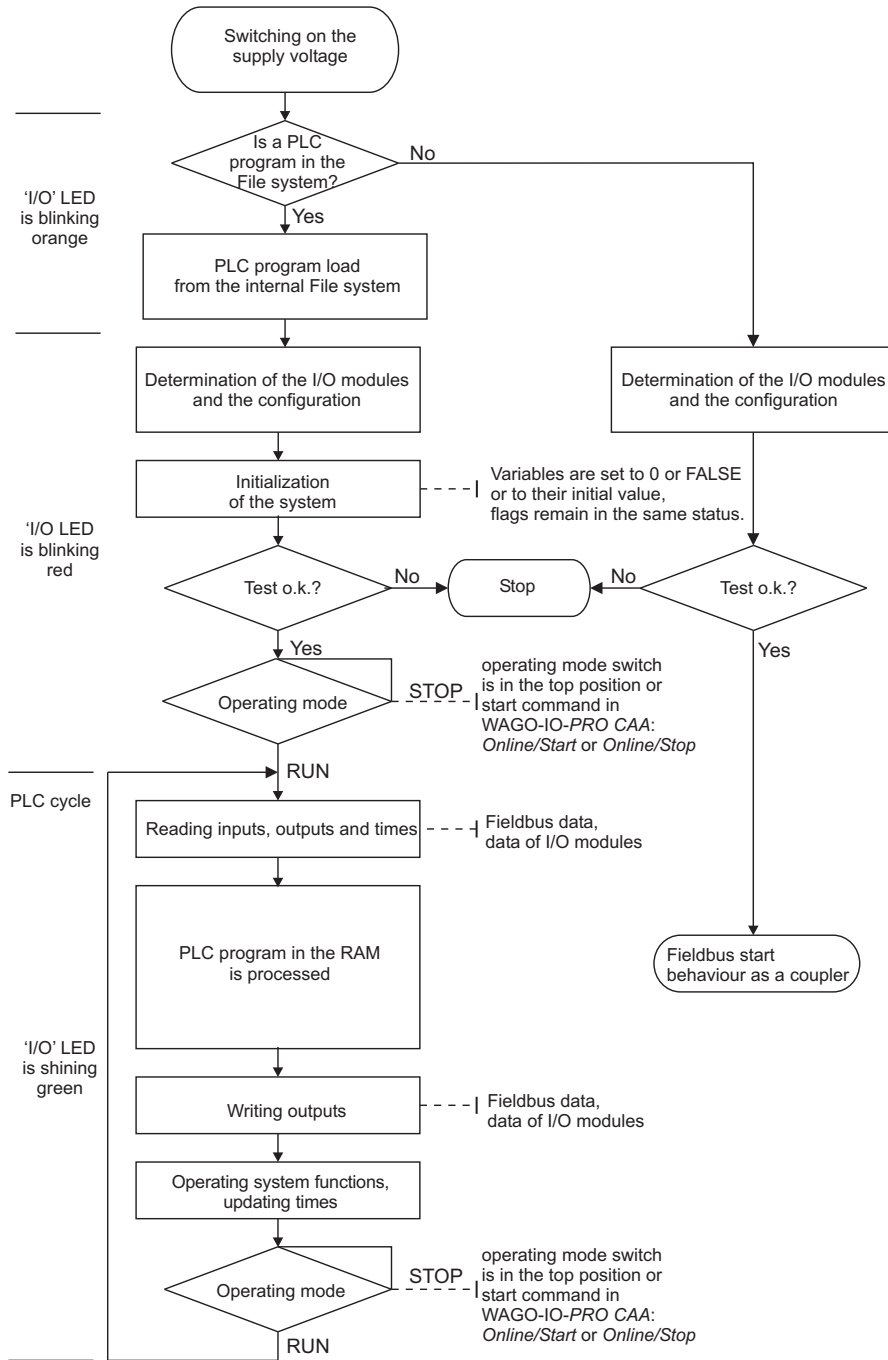


Figure 36: Run-up of the Controller

7.2 Process Data Architecture

7.2.1 Basic Structure

After switching on, the controller identifies all I/O modules connected with the node that send or receive data (data width/bit width > 0).
A node can consist of a mixed arrangement of analog and digital modules.

Note



Up to 250 I/O modules can be connected with the data bus extension modules.

Using the WAGO module bus extension coupler module 750-628 and end module 750-627 makes it possible to connect up to 250 modules to the Programmable Fieldbus Controller ETHERNET.

Information



Additional Information

For the number of input and output bits or bytes for the individual I/O modules, refer to the corresponding description of the I/O modules.

The controller creates an internal local process image on the basis of the data width, the type of I/O module and the position of the module in the node. This process image is broken down into an input and an output data range.

The data of the digital I/O modules is bit-oriented; i.e., digital data is sent bit by bit. Analog I/O modules represent the group of byte-oriented modules – data is sent byte by byte.

This group includes: counter modules, angle and distance measurement modules and communication modules.

For both the local input and the output process image, the I/O module data is stored in the corresponding process image according to the order in which the modules are connected to the controller.

Note



Hardware changes can result in changes of the process image!

If the hardware configuration is changed by adding, changing or removing of I/O modules with a data width > 0 bit, this result in a new process image structure. The process data addresses would then change. If adding modules, the process data of all previous modules has to be taken into account.

A memory range of 256 words (word 0...255) is initially available in the controller for the process image of the physical input and output data.

For the image of the MODBUS/PFC variables, the memory range of words 256...511 is reserved; meaning the image for the MODBUS/PFC variables is created behind the process image for the I/O module data.

If the quantity of module data is greater than 256 words, all the physical input and output data above this value is added to the end of the current process image in a memory range; i.e., attached behind the MODBUS/PFC variables (word 512...1275).

The Ethernet/IP PFC variables are then mapped behind the remaining physical I/O module data. This memory range includes words 1276 ... 1531.

The subsequent range, starting from word 1532, is reserved for future protocol expansion and other PFC variables.

Access by the PLC to process data is made independently from the fieldbus system in all WAGO fieldbus controllers; access is always conducted through an application-related IEC-61131-3 program.

How the data is accessed from the fieldbus side depends on the fieldbus however.

For the fieldbus controller, a MODBUS/TCP master can access the data via implemented MODBUS functions, whereby decimal or hexadecimal MODBUS addresses are used.

Optionally, data can also be accessed via Ethernet/IP using an object model.

Information



Additional Information:

For a detailed description of these fieldbus-specific data access methods, refer to the section "MODBUS Functions" or the section "Ethernet/IP (Ethernet/Industrial Protocol)".

Information



Additional Information:

For the fieldbus-specific process image of any WAGO I/O module, please refer to the section "Structure of the process data".

7.2.2 Example of an Input Process Image

The following figure is an example of an input process image. The configuration comprises 16 digital and 8 analog inputs. The input process image thus has a data length of 8 words for the analog modules and 1 word for the digital modules; i.e., 9 words in total.

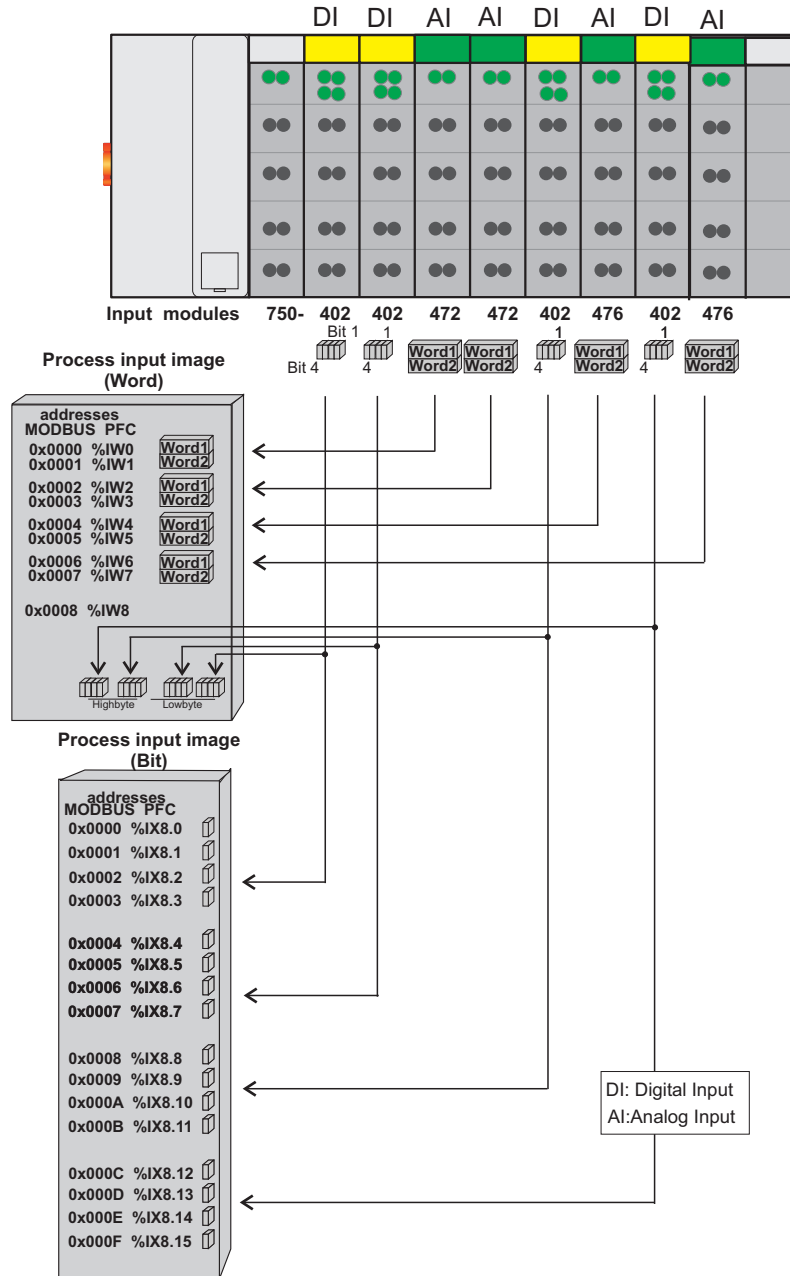


Figure 37: Example of process image for input data

7.2.3 Example of an Output Data Process Image

The following example for the output process image comprises 2 digital and 4 analog outputs. It comprises 4 words for the analog outputs and 1 word for the digital outputs; i.e., 5 words in total.

In addition, the output data can also be read back with an offset of 200_{hex} (0x0200) added to the MODBUS address.

Note



Data > 256 words can be read back by using the cumulative offset!

All output data greater than 256 words and, therefore located in the memory range 6000_{hex} (0x6000) to 66F9_{hex} (0x66F9) can be read back with an offset of 1000_{hex} (0x1000) added to the MODBUS address.

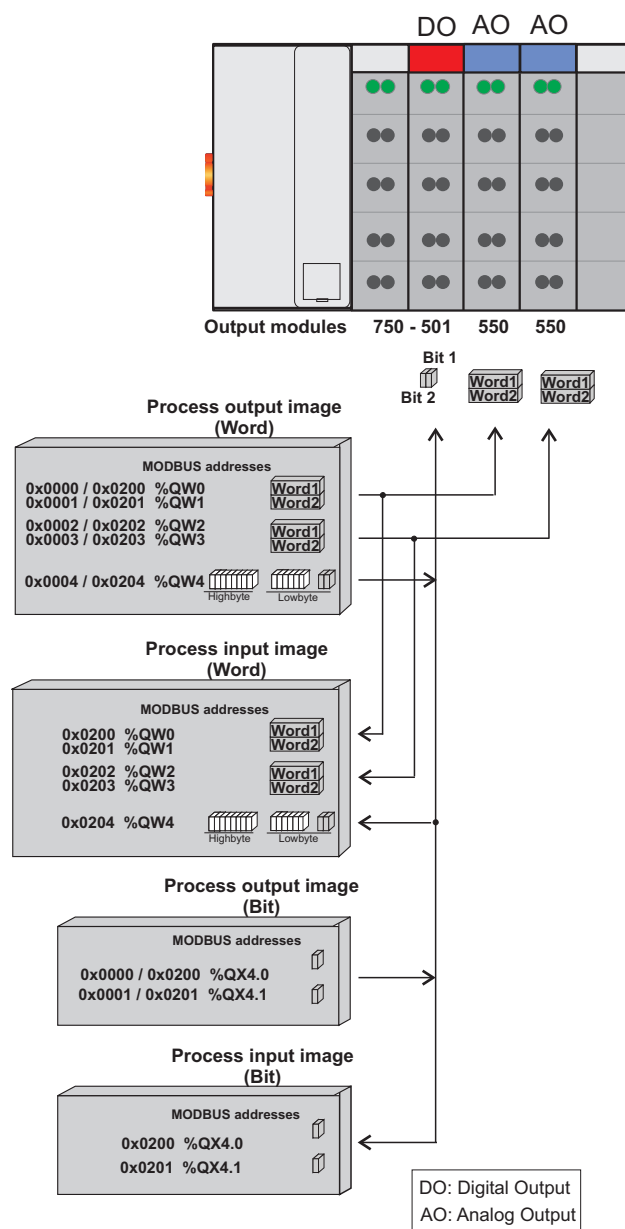


Figure 38: Example of process image for output data

7.2.4 Process Data MODBUS/TCP and EtherNet/IP

For some I/O modules (and their variations), the structure of the process data depends on the fieldbus.

For the fieldbus controller with MODBUS and Ethernet/IP, the process image is built up word-by-word (with word alignment). The internal mapping method for data greater than one byte conforms to Intel formats.

Information



Additional Information:

For the respective fieldbus-specific structure of the process values of any I/O module within the 750 or 753 Series of the WAGO-I/O-SYSTEM, refer to Section "Structure of Process Data for MODBUS/TCP" or "Structure of Process Data for Ethernet/IP".

7.3 Data Exchange

With the fieldbus controller, data is exchanged via the MODBUS/TCP protocol and/or the MODBUS/UDP protocol or Ethernet/IP.

MODBUS/TCP works according to the master/slave principle. The master controller can be a PC or a PLC.

The fieldbus controllers of the WAGO-I/O-SYSTEM 750 are usually slave devices. Thanks to the programming with IEC 61131-3, however, these controllers can also assume the master function.

The master requests communication. This request can be directed to certain nodes by addressing. The nodes receive the request and, depending on the request type, send a reply to the master.

A controller can set up a defined number of simultaneous connections (socket connections) to other network subscribers:

- 3 connections for HTTP (to read HTML pages from the controller)
- 15 connections via MODBUS/TCP (to read or write input and output data of the controller)
- 128 connections for Ethernet/IP
- 5 connections via PFC (available in the PLC function for IEC 61131-3 application programs)
- 2 connections for WAGO-I/O-PRO (these connections are reserved for debugging the application program via ETHERNET. WAGO-I/O-PRO needs 2 connections at the same time for the debugging. However, only **one** programming tool can have access to the controller).
- 10 connections for FTP
- 2 connections for SNMP

The maximum number of simultaneous connections can not be exceeded. Existing connections must first be terminated before new ones can be set up. The Programmable Fieldbus Controller ETHERNET is essentially equipped with three interfaces for data exchange:

- the interface to the fieldbus (Master),
- the PLC function of the PFC (CPU) and
- the interface to the I/O modules.

Data exchange takes place between the fieldbus master and the I/O modules, between the PLC function of the PFC (CPU) and the I/O modules and between the fieldbus master and the PLC function of the PFC (CPU).

If MODBUS is used as the fieldbus, the MODBUS master accesses the data using the MODBUS functions implemented in the controller; Ethernet/IP, in contrast, uses an object model for data access.

Data access is carried out with the aid of an IEC-61131-3 application program. Data addressing varies greatly here.

7.3.1 Memory Areas

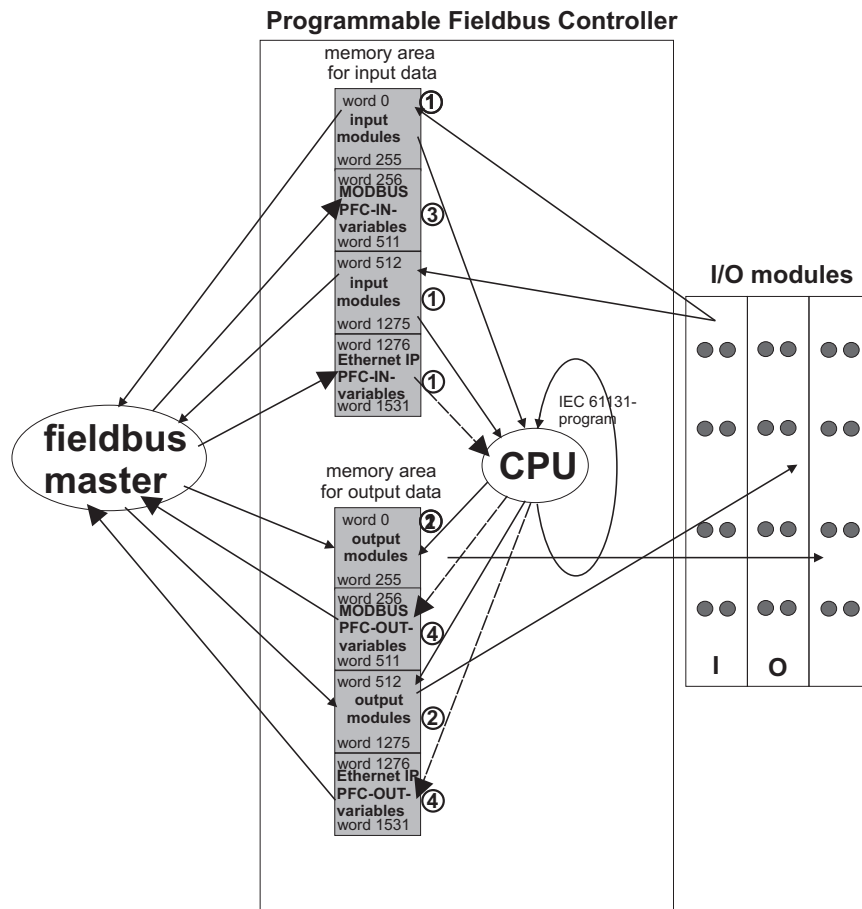


Figure 39: Memory areas and data exchange

The controller process image contains the physical data for the bus modules. These have a value of 0 ... 255 and word 512 ... 1275.

- ① The input module data can be read by the CPU and by the fieldbus side.
- ② Likewise, data can be written to the output modules from the CPU and the fieldbus side.

The MODBUS PFC variables are stored in each of the memory areas for word 256 ... 511 between these sides.

- ③ The MODBUS-PFC input variables are written to the input memory area from the fieldbus side and read in by the CPU for processing.
- ④ The variables processed by the CPU using the IEC-61131-3 program are places in the output memory area, where they can be read out by the master.

The memory area for word 1276 ... 1531 for the Ethernet/IP PFC variables is adjacent to the physical I/O module data.

The subsequent memory area, starting from word 1532, is reserved for future protocol expansion and other PFC variables.

In addition, all output data is mirrored in the Programmable Fieldbus Controller ETHERNET to a memory area with the address offset 0x0200 and 0x1000. This allows output values to be read back in by adding 0x0200 or 0x1000 to the MODBUS address.

Other memory areas are also provided in the controller, some of which cannot be accessed by the fieldbus side, however:

- **Data memory (1024 kByte)**
The data memory is a volatile RAM memory for creating variables that are not required for communication with the interfaces, but rather for internal processing procedures, such as calculation of results.
- **Program memory (1024 kByte)**
The IEC-61131-3 program is stored in the program memory. The code memory is a Flash ROM. When power is switched on, the program is transferred from the flash to the RAM memory. After error-free run-up, the PFC cycle starts with the mode selector switch at the top position, or on the Start command from the WAGO-I/O-PRO.
- **NOVRAM Remanent memory (32 kByte)**
The remanent memory is a non-volatile memory; i.e., all values of flags and variables, that are explicitly defined by “var retain”, are retained even after a loss of power. Memory management is performed automatically. The 32 kByte memory area is normally divided into an 16 kByte addressable range for flags (%MW0 ... %MW 8191) and a 16 kByte retain area for variables without memory area addressing, that are defined by "var retain".



Note

Markers are only remanent under "var retain"!

Please note that the bit memory is only retentive if you have declared it as such under "var retain".

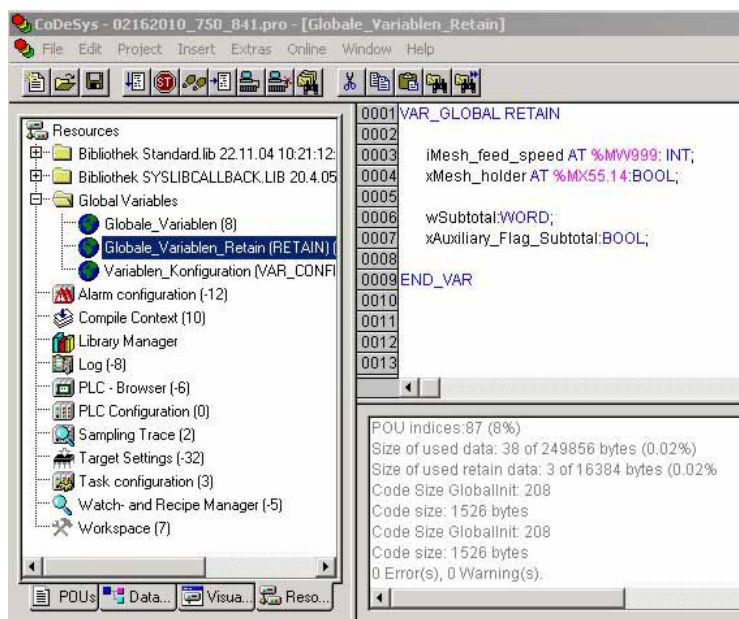


Figure 40: Example declaration of remanent flags by „var retain“

This breakdown can be varied (see following explanation).

Note



NOVRAM memory allocation can be changed in WAGO-I/O-PRO!

The breakdown of the NOVRAM can be modified when required in the programming software WAGO-I/O-PRO/Register "Resources"/Dialog window "Target system settings".

The start address for the flag area is fixed. The area sizes and the start address for the retain memory can be varied.

We do recommend keeping the standard settings, however, in order to avoid any overlapping of the areas.

In these default settings the size of the flag area is set at 16#4000, followed by the retain memory, with the size 16#4000

7.3.2 Addressing

Module inputs and outputs in a controller are addressed internally as soon as they are started. The order in which the connected modules are addressed depends on the type of module that is connected (input module, output module).

The process image is formed from these addresses.

The physical arrangement of the I/O modules in the fieldbus node is arbitrary.

Note



Use various options for addressing the bus terminals!

Connected modules in more detail. It is essential that you understand these correlations in order to conduct conventional addressing by counting.

The **WAGO I/O Configurator** is also available as a further addressing option.

The Configurator can assist you in addressing and protocol assignment for the connected modules. You must select the connected modules in the I/O Configurator; the software then takes care of correct addressing (see following Figure).

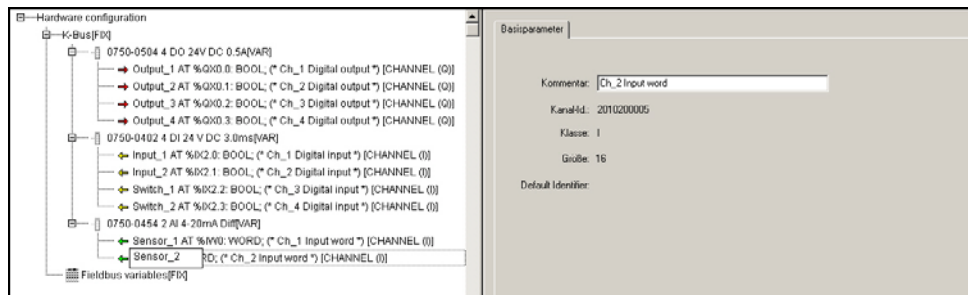


Figure 41: WAGO I/O Configurator

The I/O Configurator is started from the *WAGO-I/O-PRO*. For more details, refer to Section "Configuration using the *WAGO-I/O-PRO* I/O Configurator".

7.3.2.1 Addressing of I/O Modules

Addressing first references complex modules (modules that occupy several bytes) in accordance with their physical order downstream of the fieldbus coupler/controller; i.e., they occupy addresses starting from word 0.

Following these is the data for the remaining modules, compiled in bytes (modules that occupy less than one byte). In this process, byte by byte is filled with this data in the physical order. As soon a complete byte is occupied by the bit oriented modules, the process begins automatically with the next byte.

Note



Hardware changes can result in changes of the process image!

If the hardware configuration is changed and/or expanded; this may result in a new process image structure. In this case, the process data addresses also change. If adding modules, the process data of all previous modules has to be taken into account.

Note



Observe process data quantity!

For the number of input and output bits or bytes of the individual IO modules please refer to the corresponding description of the IO modules.

Table 26: Data with for I/O modules

Data width \geq 1 word (channel)	Data width = 1 bit (channel)
Analog input modules	Digital input modules
Analog output modules	Digital output modules
Input modules for thermocouples	Digital output modules with diagnostics (2 bits/channel)
Input modules for resistor sensors	Supply modules with fuse carrier/diagnostics
Pulse width output modules	Solid-state load relays
Interface modules	Relay output modules
Up/down counters	
I/O modules for angle and distance measurement	

7.3.2.2 Address Ranges

Subdivision of the address ranges for word-by-word addressing in accordance with IEC-61131-3:

Table 27: Breakdown of address range

Word	Data
0-255	Physical I/O modules
256-511	MODBUS PFC variables
512-1275	Other physical I/O modules
1276-1531	Ethernet/IP PFC variables
1532-...	Reserved for PFC variables with future protocols

Word 0...255: First address range for I/O module data:

Table 28: Address range Word 0...255

Data width	Address									
Bit	0.0... 0.7	0.8... 0.15	1.0... 1.7	1.8... 1.15	...	254.0... 254.7	254.8... 254.15	255.0... 255.7	255.8... 255.15	
Byte	0	1	2	3	...	508	509	510	511	
Word	0		1		...	254		255		
DWord	0				...	127				

Word 256...511: Address range for MODBUS-PFC variables:

Table 29: Address range, word 256 – 511

Data width	Address									
Bit	256..0 ...	256..8 ...	257.0 ...	257.8	510.0 ...	510.8 ...	511.0 ...	511.8 ...	
	256.7	256.15	257.7	257.15	...	510.7	510.15	511.7	511.15	
Byte	512	513	514	515	...	1020	1021	1022	1023	
Word	256		257		...	510		511		
DWord	128				...	255				

Word 512...1275: Second address range for I/O module data:

Table 30: Address range, word 512 - 1275

Data width	Address									
Bit	512.0	512.8	513.0	513.8	...	1274.0	1274.8	1275.0	1275.8	...

	512.7	512.15	513.7	513.15	...	1274.7	1274.15	1275.7	1275.15	...
Byte	1024	1025	1026	1027	...	2548	2549	2550	2551	...
Word	512		513		...	1274		1275		
DWord	256				...	637				

Word 1276-1531: Address range for Ethernet/IP fieldbus data:

Table 31: Address range, word 1276...1531

Data width	Address									
Bit	1276.0	1276.8	1277.0	1277.8	1530.0...153	1530.8...153	1531.0...153	1531.8...153
	1276.7	1276.15	1277.7	1277.15	0.7	0.15	1.7	1.15
Byte	2552	2553	2554	2555	3060	3061	3062	3063
Word	1276		1277		1530		1531		
DWord	638				765				

Address range for flags:

Table 32: Address range for flags

Data width	Address									
Bit	0.0	0.8	1.0	1.8	...	12287.0	12287.8	12288.0	12288.8	...

	0.7	0.15	1.7	1.15	...	12287.7	12287.15	12288.7	12288.15	...
Byte	0	1	2	3	...	24572	24573	24574	24575	...
Word	0		1		...	12287		12288		
DWord	0				...	6144				

IEC-61131-3 Overview of Address Areas:

Table 33: IEC-61131-3 address areas

Address area	MODBUS Access	PLC Access	Description
phys. inputs	read	read	Physical inputs (%IW0...%IW255 und %IW512...%IW1275)
phys. outputs	read/write	read/write	Physical outputs (%QW0...%QW255 und %QW512...%QW1275)
MODBUS/TCP PFC-IN variables	read/write	read	Volatile PLC input variables (%IW256...%IW511)
MODBUS/TCP PFC-OUT variables	read	read/write	Volatile PLC output variables (%QW256...%QW511)
Ethernet/IP PFC-IN variables	-	read	Volatile PLC input variables (%IW1276 ... %IW1531)
Ethernet/IP PFC-OUT variables	-	read/write	Volatile PLC output variables (%QW1276 ... %QW1531)
Configuration register	read/write	-	see Section "MODBUS Functions → MODBUS Registers → Configuration Registers"
Firmware register	read	-	see Section "MODBUS Functions → MODBUS Registers → Firmware Information Registers"
Retain variables	read/write	read/write	Remanent memory (%MW0...%MW8192)

7.3.2.3 Absolute Addressing

Direct presentation of individual memory cells (absolute addresses) based on IEC-61131-3 is performed using character strings:

Table 34: Absolute Addressing

Position	Prefix	Designation	Comment
1	%	Introduces an absolute address	
2	I Q M	Input Output Flag	
3	X* B W D	Single bit Byte (8 bits) Word (16 bits) Doubleword (32 bits)	Data width
4		Address	

such as word-by-word: %QW27 (28th word), bit-by-bit: %IX1.9 (10th bit in the 2nd word)

* The designator "X" for bits can be omitted

Note



Enter character strings without spaces or special characters!

The character strings for absolute addresses must be entered connected, i.e. without spaces or special characters!

Addressing example:

Table 35: Addressing example

	Inputs			
Bit	%IX14.0 ... 15		%IX15.0 ... 15	
Byte	%IB28	%IB29	%IB30	%IB31
Word	%IW14		%IW15	
Double word	%ID7			
	Outputs			
Bit	%QX5.0 ... 15		%QX6.0 ... 15	
Byte	%QB10	%QB11	%QB12	%QB13
Word	%QW5		%QW6	
Double word	%QD2 (top section)		%QD3 (bottom section)	

Flags			
Bit	%MX11.0 ... 15		%MX12.0 ... 15
Byte	%MB22	%MB23	%MB24 %MB25
Word	%MW11		%MW12
Double word	%MD5 (top section)		%MD6 (bottom section)

Calculating addresses (as a function of the word address):

Bit address: Word address .0 to .15

Byte address: 1st byte: 2 x word address
 2nd byte: 2 x word address + 1

DWord address: Word address (even number) / 2
 or Word address (uneven number) / 2, rounded down

7.3.3 Data Exchange between MODBUS/TCP Master and I/O Modules

Data exchange between the MODBUS/TCP Master and the I/O modules is conducted using the MODBUS functions implemented in the controller by means of bit-by-bit or word-by-word reading and writing routines.

There are 4 different types of process data in the controller:

- Input words
- Output words
- Input bits
- Output bits

Access by word to the digital I/O modules is carried out in accordance with the following table:

Table 36: Allocation of digital inputs and outputs to process data words in accordance with the Intel format

Digital inputs/ outputs	16.	15.	14.	13.	12.	11.	10.	9.	8.	7.	6.	5.	4.	3.	2.	1.
Process data word	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte	High byte D1								Low byte D0							

Output can be read back in by adding an offset of 200_{hex} (0x0200) to the MODBUS address.

Note



Data > 256 words can be read back by using the cumulative offset!

All output data greater than 256 words and, therefore located in the memory range 0x6000 to 0x62FC, can be read back by adding an offset of 1000_{hex} (0x1000) to the MODBUS address.

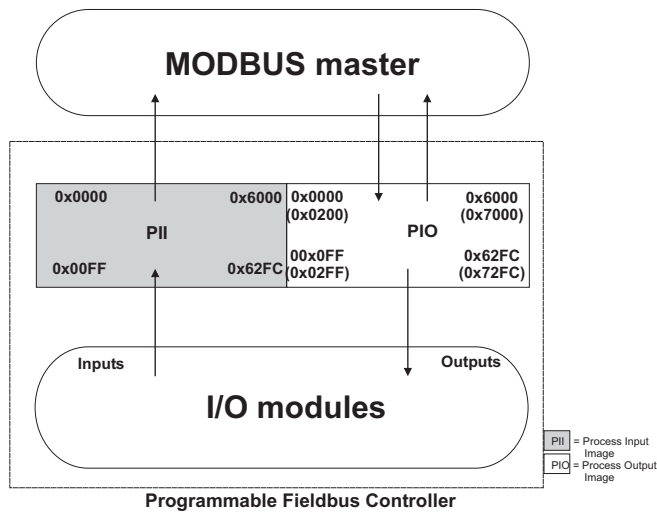


Figure 42: Data exchange between MODBUS Master and I/O modules

Register functions start at address 0x1000. These functions can be addressed in a similar manner with the MODBUS function codes that are implemented (read/write).

The specific register address is then specified instead of the address for a module channel.

Information



Additional Information

A detailed description of the MODBUS addressing may be found in Chapter "MODBUS Register Mapping".

7.3.3.1 Data Exchange between EtherNet/IP Master and I/O Modules

The data exchange between Ethernet/IP master and the I/O modules is object-oriented. Each node on the network is depicted as a collection of objects.

The "assembly" object specifies the structure of the objects for the data transmission. With the assembly object, data (e.g. I/O data) can be combined into blocks (mapped) and sent via a single message connection. Thanks to this mapping, less access to the network is necessary.

There is a distinction between input and output assemblies.

An input assembly reads in data from the application via the network or produces data on the network.

An output assembly writes data to the application or consumes data from the network.

In the fieldbus coupler/controller, various assembly instances are already preprogrammed (static assembly).

After the input voltage is applied, the assembly object combines data from the process image. As soon as a connection is established, the master can address the data with "class", "instance", and "attribute" and access it or read and write using I/O connections.

The mapping of the data depends on the assembly instance of the static assembly selected.

Information



Additional Information:

The assembly instances for the static assembly are described in the section "Ethernet/IP".

7.3.4 Data Exchange between PLC Function (CPU) and I/O Modules

The PLC function (CPU) of the PFC uses direct addresses to access the I/O module data.

The PFC uses absolute addresses to reference the input data. The data can then be processed internally in the controller using the IEC-61131-3 program. Flags are stored in a non-volatile memory area in this process. The results of linking can then be written directly to the output data employing absolute addressing.

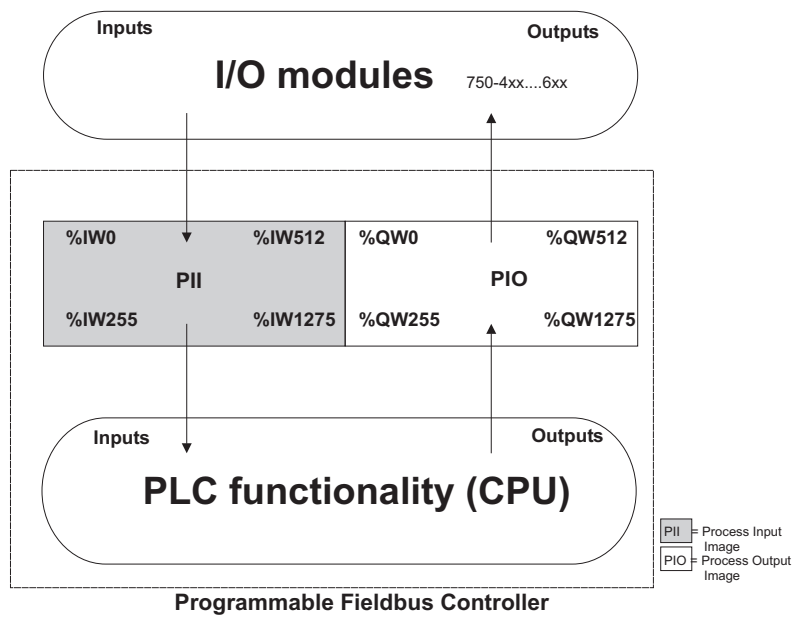


Figure 43: Data exchange between PLC function (CPU) of the PFC and the I/O modules

7.3.5 Data Exchange between Master and PLC Function (CPU)

The fieldbus master and the PLC function (CPU) of the PFC have different perspectives on data.

Variable data generated by the master are routed as input variables to the PFC, where they are further processed.

Data created in the PFC are transmitted via fieldbus to the master as output variables.

In the PFC, access to the MODBUS/TCP PFC variable data is possible starting from word address 256 to 511 (double-word address 128-255, byte address 512-1023), while access to the PFC variable data is possible starting from a word address of 1276 to 1531 (double-word address 638-765, byte address 2552-3063).

7.3.5.1 Example of MODBUS/TCP Master and PLC Function (CPU)

Data access by the MODBUS/TCP Master

Access to data by the MODBUS Master is always either by word or by bit. Addressing of the first 256 data words by the I/O modules begins with word-by-word and bit-by-bit access at 0.

Addressing of the data by the variables begins at 256 for word-based access; bit-by-bit access then takes place starting at:

4096 for bit 0 in word 256
4097 for bit 1 in word 256
...
8191 for bit 15 in word 511.

The bit number can be determined using the following equation:

$$\text{Bit No.} = (\text{word} * 16) + \text{Bit No. in word}$$

Example: 4097 = (256 * 16) + 1

Data Access by PLC Function (CPU)

The PLC function of the PFC employs a different type of addressing for accessing the same data. PLC addressing is identical with word-by-word addressing by the MODBUS Master for the declaration of 16-bit variables. However, a different notation is used for declaration of Boolean variables (1 bit) than that used by MODBUS. Here, the bit address is composed of the elements word address and bit number in the word, separated by a decimal point.

Example:

Bit access by MODBUS to bit number 4097 => Bit addressing in the PLC
<Word No.>.<Bit No.> = 256.1

The PLC function of the PFC can also access data by bytes and by doubleword access.

Addresses are calculated based on the following equations for byte-based access:

$$\begin{aligned}\text{High Byte address} &= \text{Word address} * 2 \\ \text{Low Byte address} &= (\text{Word address} * 2) + 1\end{aligned}$$

Addresses are calculated according to the following equation for double-word based access:

$$\begin{aligned}\text{Double-word address} &= \text{High word address} / 2 \text{ (rounded down)} \\ &\text{or} = \text{Low word address} / 2\end{aligned}$$

Information



Additional Information

There is a detailed description of the MODBUS and the corresponding IEC 61131 addressing in section "MODBUS Register Mapping".

7.3.6 Application Example

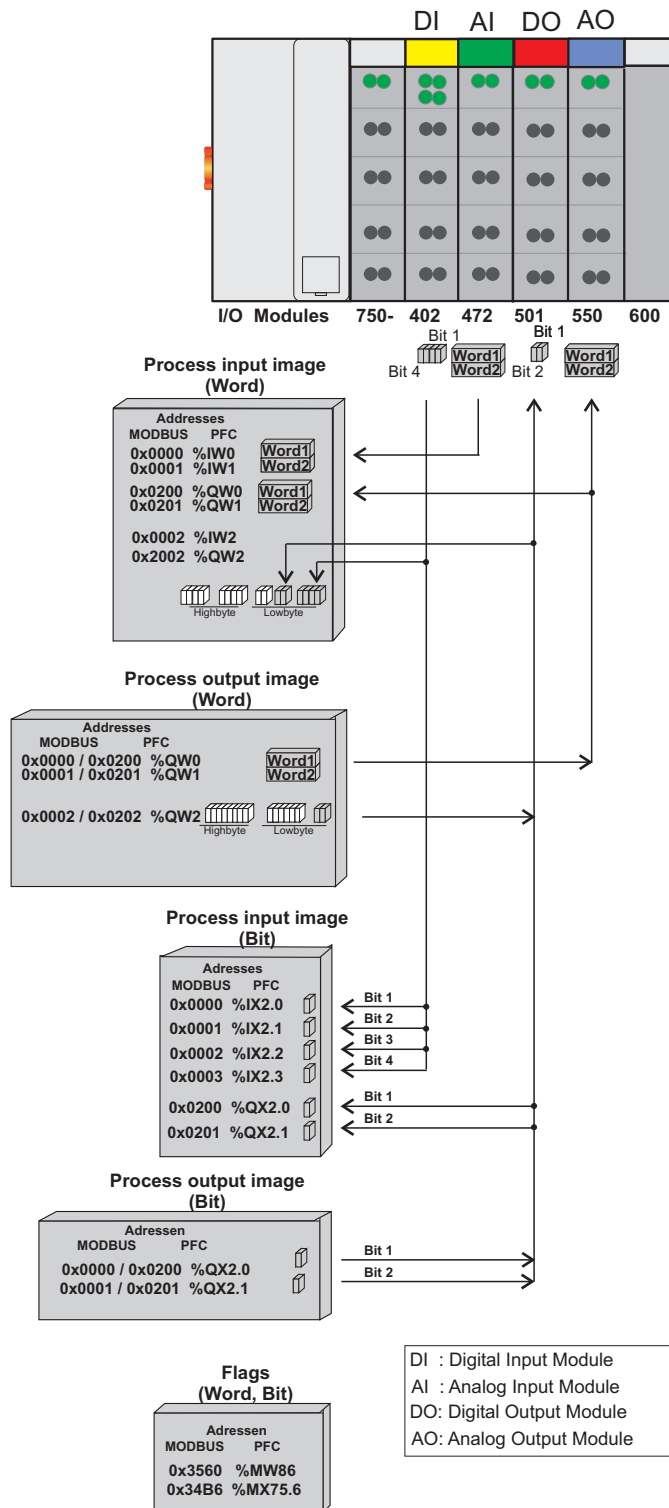


Figure 44: Example of addressing for a fieldbus node

8 Commissioning

This chapter shows a step-by-step procedure for starting up exemplarily a WAGO fieldbus node.

Note



Good example!

This description is just an example and only serves to describe the procedure for a local start-up of a single fieldbus node with a non-networked computer under Windows.

Two work steps are required for start-up. The description of these work steps can be found in the corresponding following sections.

- **Connecting client PC and fieldbus nodes**
- **Assigning the IP address to the fieldbus node**

Note



The IP address must occur in the network only once!

For error-free network communication, note that the assigned IP address must occur only once in the network!

In the event of an error, the error message "IP address configuration error" (error code 6 - error argument 6) is indicated by 'I/O' LED at the power-on.

There are various ways to assign the IP address.

The various options are described in the following sections individually.

Following the commissioning descriptions after which the fieldbus node is ready for communication, the following topics are described:

- **Preparing the Flash File System**
- **Synchronizing the real-time clock**
- **Restoring factory settings**

After the topics specified above, you can find instructions for programming the fieldbus controller with WAGO-I/O-PRO and the description of the internal web pages of the web-based Management System (WBM) for additional settings of the fieldbus controller.

8.1 Connecting Client PC and Fieldbus Nodes

1. Mount the fieldbus to the carrier rail.
Observe the installation instructions described in "Assembly" section.
2. Connect the 24V power supply to the supply module.
3. Connect an Ethernet interface from the client PC to an Ethernet interface of the fieldbus controller
4. Turn the operating voltage on.
Make sure that the mode selector is not in the bottom position.

After the power is switched on, the controller is initialized. The fieldbus controller determines the I/O module configuration and creates a process image. During startup, the I/O LED (red) will flash. After a brief period, the I/O LED lights up green, indicating that the fieldbus controller is operational.

If an error has occurred during startup, a fault code is flashed on the I/O LED. If the I/O LED flashes 6 times (indicating error code 6) and then 4 times (indicating error argument 4), an IP address has not been assigned yet.

8.2 Allocating the IP Address to the Fieldbus Node

- Use **address selection switch** (DIP switch) to assign IP address (manually).
- **Automatic assignment of addresses via DHCP**
- **Assigning IP Address via BootP server**

8.2.1 Assigning IP Address via Address Selection Switch

Use the address selection switch to set the host ID, i.e., the last byte of the IP address, which is entered in the Web-Based Management System on WBM page “TCP/IP”, entry “DIP switch IP-Adress”, with values between 1 and 254 binary coded.

Example:

DIP switch IP address: 192.168.7

Set DIP switch value: **50** (binary coded: 00110010)

Resulting IP address saved in the fieldbus controller: 192.168.7.**50**

Note



Host ID 1 - 254 via address selection switch freely adjustable!

Use the address selection switch to set the last byte of the IP address to a value between 1 and 254. The DIP switch is then enabled and the IP address is composed of the DIP switch base address stored in the fieldbus controller and the host ID set on the DIP switch.

The IP address make via the Web-based Management is disabled.

Note



Address selection switch values 0 and 255 are predefined, address selection switch disabled!

If you use the address selection switch to set the value 0 or 255, the address selection switch is disabled and the setting configured in the fieldbus controller is used.

With the value 0, the settings of the Web based Management System apply.

If you set the value 255, the configuration via DHCP is activated.

The base address used consists of the first three bytes of the IP address. This always depends on the DIP switch IP address currently saved in the fieldbus controller.

If there are still no static IP address in the fieldbus controller, the default value **192.168.1** defined by the firmware as the base address is used when setting the DIP switch to 1 - 254.

The address selection switch setting then overwrites the value of the host ID.

Information



More information about changing the static base address

You can also change the base address currently saved in the fieldbus controller as required.

Either proceed as described in the following section "Assigning IP Address via Web Server".

1. To configure the IP address via the address selection switch by setting the host ID (last position of the IP address) to a value that does not equal 0/255, first convert the host ID to the binary representation.
For example, host ID 50 results in a binary code of 00110010.
2. Set the bits in sequence using the 8 address switches. Start with address switch 1 to set bit 0 (LSB) and end with address switch 8 for bit 7 (MSB).

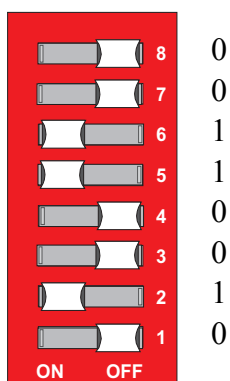


Figure 45: Address selection switch

3. Restart the fieldbus coupler after adjusting the address selection switch to apply the configuration changes.

8.2.2 Assigning IP Address via DHCP

If you want to use DHCP to assign the IP address, it happens automatically via a DHCP server on the network.

Note

**Total network failure when there are two DHCP servers in the network!**

To prevent network failure, never connect a PC, on which a DHCP server is installed, to a global network. In larger networks, there is usually a DHCP server already that can cause collisions and subsequent network failure.

Note

**There must be a DHCP server in the network for further configuration!**

Install a DHCP server on your client PC in the local network if not already available. You can download a DHCP server free of charge on the Internet, e.g., http://windowspedia.de/dhcp-server_download/.

Note

**Assign the client PC a fixed IP address and note common subnet!**

Note that the client PC, on which the DHCP server is listed, must have a fixed IP address and that the fieldbus node and client PC must be in the same subnet.

The following steps are included:

- Enable DHCP
- Disable DHCP

8.2.2.1 Enable DHCP

Note

**Set the address selection switch to 255 for active software configuration!**

Set the address selection switch to 255 to disable the DIP switch and to enable DHCP.

Restart the fieldbus node after adjusting the address selection switch to apply the configuration changes.

8.2.2.2 Disabling DHCP

Note



BootP must be disabled to assign the address permanently!

To apply the new IP address permanently in the fieldbus controller, BootP must be disabled.

This prevents the fieldbus coupler from receiving a new BootP request.

You can disable DHCP in two ways:

- Disable DHCP via the address selection switch.
- Disable DHCP in the Web-based Management System.

Disable DHCP via the address selection switch.

Note



Do not set the address selection switch to 0/255 again!

Do not switch the address selection switch to 0/255 again because doing so automatically disables the DIP switch and enables IP address assignment via the software configuration.

1. Use the address selection switch to set a value between 1 ... 254 and the DIP switch IP address saved in the fieldbus controller (with changed Host ID = DIP switch) is then valid.
(Example: If the DIP switch IP address 10.127.3 was saved in the fieldbus controller and you set the switch to 50 (binary coded 00110010), for example, the fieldbus controller then has the address 10.127.3.50.)
2. Restart the fieldbus coupler after adjusting the address selection switch to apply the configuration changes.

Disable DHCP in the Web-based Management System

Note



Set the address selection switch to 0 for active software configuration!

Set the address selection switch to 0 to disable address selection via DIP switch or DHCP.

1. Set the address selection switch to 0.
2. Launch a Web browser (e.g., MS Internet Explorer or Mozilla) and enter the IP address you have assigned your fieldbus node in the address bar.

3. Click **[Enter]** to confirm.
The start page of the Web based Management System loads.
4. Select "Port" in the left menu bar.
5. Enter your user name and password in the inquiry screen (default: user = "admin", password = "wago" or user = "user", password = "user").
The HTML page "Port configuration" loads:

WAGO INNOVATIVE CONNECTIONS

Web-based Management

WAGO Kontakttechnik GmbH & Co. KG
Hansastr. 27
D-32423 Minden
www.wago.com

Navigation

- Information
- Ethernet
- TCP/IP
- Port**
- SNMP
- SNMP V3
- Watchdog
- Clock
- Security
- PLC
- Features
- IO config
- WebVisu

Port configuration

This page is for the configuration of the network protocols. The configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

Port Settings

Protocol	Port	Enabled
FTP	21	<input checked="" type="checkbox"/>
SNTP	123	<input type="checkbox"/>
HTTP	80	<input checked="" type="checkbox"/>
SNMP	161, 162	<input type="checkbox"/>
Ethernet IP	44818 (TCP), 2222 (UDP)	<input checked="" type="checkbox"/>
Modbus UDP	502	<input checked="" type="checkbox"/>
Modbus TCP	502	<input checked="" type="checkbox"/>
WAGO Services	6626	<input checked="" type="checkbox"/>
CoDeSys	2455	<input checked="" type="checkbox"/>
BootP	68	<input type="radio"/>
DHCP	68	<input type="radio"/>
use IP from EEPROM	--	<input checked="" type="radio"/>

UNDO SUBMIT

Figure 46: WBM page "Port"

6. Disable DHCP by selecting the option "**BootP**" or "**use IP fom EEPROM**".
7. Click on **[SUBMIT]** to apply the changes in your fieldbus node.
8. Restart the fieldbus node to apply the settings of the Web interface.

8.2.3 Assigning the IP Address with a BootP Server

A BootP server or PLC program can be used to assign a fixed IP address.

When assigning an address using a PLC program, this can be done using the "Ethernet_Set_Network_Config" function block from the "Ethernet.lib" library integrated in WAGO-I/O-PRO.

Assigning the IP address using a BootP server depends on the respective BootP program. Handling is described in the respective manual for the program or in the respective integrated help texts.

Note



Set the address selection switch to 0 for active software configuration!

Set the address selection switch to 0 to disable the DIP switch and to enable the software configuration via BootP.

Restart the fieldbus node after adjusting the address selection switch to apply the configuration changes.

Note



IP address assignment is not possible via the router!

The IP address is assigned via patch cable, switches, hubs, or via direct link using a crossover cable. Addresses can not be allocated via router.

Note



BootP must be enabled on the Web pages!

Note that BootP must be enabled on the internal Web pages of the WBM, HTML page "Port configuration".

BootP is enabled by default when delivered.

Information



Additional Information

Assigning IP addresses using the WAGO-BootP server can be carried out in any Windows and Linux operating system. Any other BootP servers may also be used, besides the WAGO-BootP server.

Information



More information about the WAGO-BootP-Server

The "WAGO-BootP-Server 759-315" is available free of charge on the CD "AUTOMATION Tools and Docs" (Art. No.: 0888-0412) or at <http://www.wago.com> under Downloads → AUTOMATION → 759-315 WAGO-BootP-Server.

The following steps are included:

- Note MAC ID
- Note IP address
- Assigning the IP address and enable BootP
- Disable BootP

8.2.3.1 Note MAC ID

1. Write down the controller's MAC address (see label or peel-off strip).
If the fieldbus is already installed, turn off the operating voltage of the fieldbus controller, then take the fieldbus controller out of the assembly of your fieldbus node and note the MAC ID of your fieldbus controller.

The MAC ID is applied to the back of the fieldbus controller or on the self-adhesive peel-off strip on the side of the fieldbus controller.

MAC ID of the fieldbus controller: `00:30:DE:__:__:__`

2. Plug the fieldbus controller into the assembly of the fieldbus node.
3. Use the fieldbus cable to connect the fieldbus connection of your mechanically and electrically assembled fieldbus node to an open interface on your computer.
The client PC must be equipped with a network card for this connection.
The controller transfer rate then depends on the network card of your client PC.
4. Start the client that assumes the function of the master and BootP server.
5. Switch on the power at the controller (DC 24 V power supply unit).

After the power is switched on, the controller is initialized. The fieldbus controller determines the I/O module configuration and creates a process image. During startup, the I/O LED (red) will flash. After a brief period, the I/O LED lights up green, indicating that the fieldbus controller is operational.

If an error occurs during start-up indicated by the I/O LED flashing red, evaluate the error code and argument and resolve the error.



Information

More information about LED signaling

The exact description for evaluating the LED signal displayed is available in the section "Diagnostics", "LED Signaling".

Error codes and error arguments are indicated by the frequency of a LED flash sequence. For example: Error code 6, followed by error argument 4, is indicated by the I/O LED after controller start-up with 6 red error code flashes, followed by four red flashes of the error argument. This indicates that an IP address has not yet been assigned.

8.2.3.2 Determining IP addresses

1. If the client PC is already integrated into an IP network, you can determine the client PC's IP address by clicking on **Control Panel** from the **Start Menu / Settings**.
2. Double-click on the **Network** icon.
The network dialog window appears.

For Windows NT:

- Select the **Protocols** tab
- Mark the entry TCP/IP protocol

For Windows 2000/XP:

- Select Network and Dial-Up Connections
- In the dialog window that then appears, right click on **LAN Connection** and open the Properties link.
- Mark the entry **Internet Protocol (TCP/IP)**

Note



Reinstall TCP/IP components if required!

If the "Internet Protocol TCP/IP" entry is missing, install the corresponding TCP/IP components and reboot your computer.

You will need the installation CD for Windows NT, 2000 or XP.

3. Then click on the **Properties...** button
4. The IP address, subnet mask and, where required, the client PC's gateway address appear in the Properties window. Note these values:

Client PC IP address:

Subnet mask:

Gateway:

____ . ____ . ____ . ____
____ . ____ . ____ . ____
____ . ____ . ____ . ____

5. Now select the desired IP address for your fieldbus node.

Note



Assign the client PC a fixed IP address and note common subnet!

Note that the client PC, on which the BootP server is listed, must have a fixed IP address and that the fieldbus node and client PC must be in the same subnet.

6. Note the IP address you have selected:

Fieldbus node IP address:

____ . ____ . ____ . ____

8.2.3.3 Assigning the IP address and Enable BootP

1. Based on the handling, which depends on the BootP program set, assign the required IP address for your fieldbus node.
2. Enable the query/response mechanism of the BootP protocol based on the handling, which depends on the BootP program set.
3. To apply the new IP address, use a hardware reset to restart your fieldbus node (interrupt the voltage supply for approx. 2 seconds).

8.2.3.4 Disabling BootP

When the BootP protocol is activated the controller expects the BootP server to be permanently available. If there is no BootP server available after a PowerOn reset, the network will remain inactive.

You must then deactivate the BootP protocol so that the controller uses the configured IP address from the EEPROM; this does away with the need for the BootP server to be permanently available.

Note



BootP must be disabled to assign the address permanently!

To apply the new IP address permanently in the fieldbus controller, BootP must be disabled.

This prevents the fieldbus coupler from receiving a new BootP request.

Note



The IP address is not lost when the BootP-Protocol is disabled!

If the BootP protocol is deactivated after addresses have been assigned, the stored IP address is retained, even after an extended loss of power, or when the controller is removed

You can disable in the Web-based Management System.

Disable BootP in the Web-based Management System

1. Open the **Web browser** on your client (such as the Microsoft Internet Explorer) to have the HTML pages displayed.
2. Enter the **IP address** for your fieldbus node in the address line of the browser and press [**Return**].

A dialog window then appears with a password prompt. This is provided for secure access and entails three different user groups: admin, guest and user.

- As Administrator, enter the user name: "**admin**" and the password "**wago**".

A start page is then displayed in the browser window with information about your fieldbus controller. You can navigate to other information using the hyperlinks in the left navigation bar.

WAGO INNOVATIVE CONNECTIONS

Web-based Management

WAGO Kontakttechnik GmbH & Co. KG
Hansstr. 27
D-32423 Minden
www.wago.com

Navigation

- Information
- Ethernet
- TCP/IP
- Port
- SNMP
- SNMP V3
- Watchdog
- Clock
- Security
- PLC
- Features
- IO config
- WebVisu

Status information

Coupler details

Order number	750-881
Mac address	0030DE038178
Firmware revision	01.01.13 (01)

Actual network settings

IP address	192.168.1.206
	Static Configuration
Subnet mask	255.255.0.0
Gateway	0.0.0.0
Hostname	
Domainname	
(S)NTP-Server	0.0.0.0
DNS-Server 1	0.0.0.0
DNS-Server 2	0.0.0.0

Module status

State Modbus Watchdog:	Disabled
Error code:	0
Error argument:	0
Error description:	Coupler running, OK

Figure 47: WBM page "Information"



Note

Disable the proxy server to display the web-based Management-System!

If these pages are not displayed for local access to the fieldbus nodes, you must define in the Web browser properties that, as an exception, no proxy server are to be used for the node IP address.



Note

The controller IP can be changed in the network by the DHCP server!

If DHCP is activated and an DHCP server is installed in the network (e.g. a router with DHCP server activated) addresses will be assigned automatically from the address range for the router after a loss of power (loss of 24 V DC power to controller). As a result, all controllers with activated DHCP will be assigned new IP addresses!

- In the left navigation bar click on **Port** to open the HTML page for selecting a protocol.

WAGO INNOVATIVE CONNECTIONS

Web-based Management

WAGO Kontakttechnik GmbH & Co. KG
Hansastr. 27
D-32423 Minden
www.wago.com

Navigation

- Information
- Ethernet
- TCP/IP
- Port**
- SNMP
- SNMP V3
- Watchdog
- Clock
- Security
- PLC
- Features
- IO config
- WebVisu

Port configuration

This page is for the configuration of the network protocols. The configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

Port Settings

Protocol	Port	Enabled
FTP	21	<input checked="" type="checkbox"/>
SNTP	123	<input type="checkbox"/>
HTTP	80	<input checked="" type="checkbox"/>
SNMP	161, 162	<input type="checkbox"/>
Ethernet IP	44818 (TCP), 2222 (UDP)	<input checked="" type="checkbox"/>
Modbus UDP	502	<input checked="" type="checkbox"/>
Modbus TCP	502	<input checked="" type="checkbox"/>
WAGO Services	6626	<input checked="" type="checkbox"/>
CoDeSys	2455	<input checked="" type="checkbox"/>
BootP	68	<input type="radio"/>
DHCP	68	<input type="radio"/>
use IP from EEPROM	--	<input checked="" type="radio"/>

UNDO SUBMIT

Figure 48: WBM page "Port"

You are shown a list of all the protocols supported by the controller.

- Select the option "**DHCP**" or "**use IP from EEPROM**".
You have now deactivated the BootP protocol.

You can also deactivate any other protocols that you no longer need in the same manner, or select desired protocols and activate them explicitly.

Since communication for each protocol takes place via different ports, you can have several protocols activated simultaneously; communication takes place via these protocols.

6. Click on **SUBMIT** and then switch off the power to the controller (hardware reset), or press down the mode selector switch. The protocol settings are then saved and the controller is ready for operation.

If you have activated the MODBUS/TCP protocol, for example, you can now select and execute required MODBUS functions using the MODBUS master too, such as querying of the module configuration via register 0x2030.

If you have activated the WAGO-I/O-PRO for example, you can also program the controller via ETHERNET link using WAGO-I/O-PRO in line with Standard IEC-61131-3.

8.2.3.5 Reasons for Failed IP Address Assignment

- The controller MAC address does not correspond to the entry given in the "bootstrap.txt" file.
- The client on whom the BootP server is running is not located in the same subnet as the controller; i.e., the IP addresses do not match
Example: Client IP: 192.168.0.10 and controller IP: 10.1.254.5
- Client and/or controller is/are not linked to the ETHERNET
- Poor signal quality (use switches or hubs)

8.3 Testing the Function of the Fieldbus Node

1. To ensure that the IP address is correct and to test communication with the fieldbus node, first turn off the operating voltage of the fieldbus node.
2. Create a non-serial connection between your client PC and the fieldbus node.

After the power is switched on, the controller is initialized. The fieldbus controller determines the I/O module configuration and creates a process image. During startup, the I/O LED (red) will flash. After a brief period, the I/O LED lights up green, indicating that the fieldbus controller is operational.

If an error occurs during start-up indicated by the I/O LED flashing red, evaluate the error code and argument and resolve the error.



Information

More information about LED signaling

The exact description for evaluating the LED signal displayed is available in the section "Diagnostics", "LED Signaling".

3. To test the coupler's newly assigned I/P address, start a DOS window by clicking on the **Start** menu item **Programs/MS-DOS Prompt**.
4. In the DOS window, enter the command: "**ping**" followed by the IP address of your coupler in the following format:

`ping [space] XXX . XXX . XXX . XXX (=IP address)`

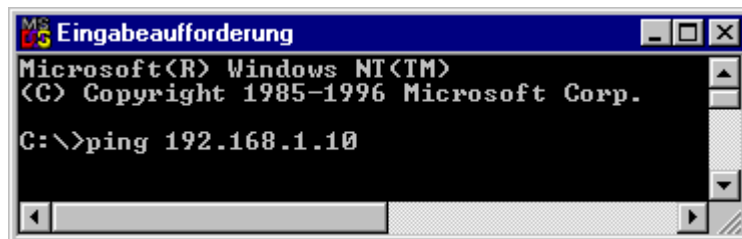


Figure 49: Example for the Function test of a Fieldbus Node

5. When the **[Enter]** key has been pressed, your PC will receive a query from the coupler, which will then be displayed in the DOS window. If the error message: "Timeout" appears, please compare your entries again to the allocated IP address and check all connections.
6. When the test has been performed successfully, you can close the DOS prompt.

The fieldbus node is now ready for communication.

8.4 Preparing the Flash File System

The flash file system must be prepared in order to use the Web interface of the fieldbus controller to make all configurations.

The flash file system is already prepared when delivered.

However, if the flash file system has not been initialized on your fieldbus controller or it has been destroyed due to an error, you must first extract it to the flash memory to access it.

NOTICE

Do not connect 750-920 Communication Cable when energized!

To prevent damage to the communications interface, do not connect or disconnect 750-920 Communication Cable when energized! The fieldbus coupler must be de-energized!

Note



Formatting erases data!

Note that formatting erases all data and configurations.

Only use this function when the flash file system has not been initialized yet or has been destroyed due to an error.

1. Switch off the supply voltage of the fieldbus controller.
2. Connect the communication cable 750-920 to the configuration interface of the fieldbus controller and to a vacant serial port on your computer.
3. Switch on the supply voltage of the fieldbus controller.

After the power is switched on, the controller is initialized. The fieldbus controller determines the I/O module configuration and creates a process image. During startup, the I/O LED (red) will flash. After a brief period, the I/O LED lights up green, indicating that the fieldbus controller is operational.

If an error occurs during start-up indicated by the I/O LED flashing red, evaluate the error code and argument and resolve the error.

Information



More information about LED signaling

The exact description for evaluating the LED signal displayed is available in the section "Diagnostics", "LED Signaling".

4. Start the **WAGO-ETHERNET-Settings** program.

5. In the top menu bar, select **Format** to format the file system.
Formatting is complete when the status window displays "Formatting flash disk successfully done".
6. In the top menu bar, select **Extract** to extract the Web pages of the flash file system.
This process takes a few seconds and is complete when the status window displays "Extracting files successfully done."

Note



Restart the Fieldbus coupler/controller after [Format]/[Extract]!

Make a restart of the fieldbus coupler/controller, so that the Web pages can be displayed after a Format/Extract.

8.5 Synchronizing the Real-Time Clock

The fieldbus controller's real-time clock enables a date and time indication for files in the flash file system.

At start-up, synchronize the real-time clock with the computer's current time.

There are two options to synchronize the real-time clock:

- Synchronize the real-time clock using **WAGO-ETHERNET-Settings**
- Synchronize the real-time clock using the **Web-based Management-System**

Synchronize the real-time clock using WAGO-ETHERNET-Settings

1. Switch off the supply voltage of the fieldbus controller.
2. Connect the communication cable 750-920 to the configuration interface of the fieldbus controller and to a vacant serial port on your computer.
3. Switch on the supply voltage of the fieldbus controller.

After the power is switched on, the controller is initialized. The fieldbus controller determines the I/O module configuration and creates a process image. During startup, the I/O LED (red) will flash. After a brief period, the I/O LED lights up green, indicating that the fieldbus controller is operational.

If an error occurs during start-up indicated by the I/O LED flashing red, evaluate the error code and argument and resolve the error.



Information

More information about LED signaling

The exact description for evaluating the LED signal displayed is available in the section "Diagnostics", "LED Signaling".

4. Start the **WAGO Ethernet Settings** program.
5. Select the **Real-time Clock** tab.

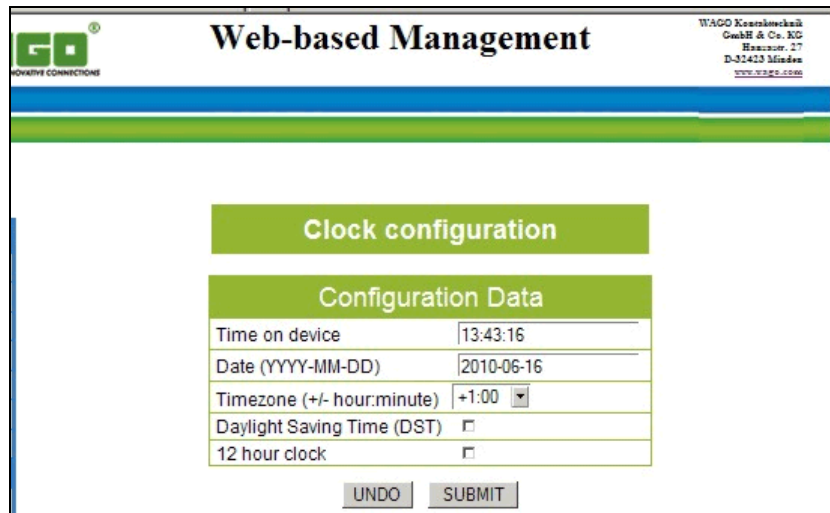


Figure 50: Example of real-time clock synchronization in ETHERNET Settings

6. Click on the "Synchronize" button with the clock icon.

Synchronize the real-time clock using the Web-based Management-System

1. Launch a Web browser (e.g., MS Internet Explorer or Mozilla) and enter the IP address you have assigned your fieldbus node in the address bar.
2. Click **[Enter]** to confirm.
The start page of the Web interface loads.
3. Select "Clock" in the left menu bar.
4. Enter your user name and password in the inquiry screen (default: user = "admin", password = "wago" or user = "user", password = "user").
The HTML page "Clock configuration" loads:



The screenshot displays the 'Web-based Management' interface for a WAGO device. At the top left is the 'GO' logo with 'POSITIVE CONNECTIONS' below it. At the top right is the company information: 'WAGO Kontakttechnik GmbH & Co. KG, Heisenstr. 27, D-31423 Minden, www.wago.com'. The main content area features a green header 'Clock configuration' and a 'Configuration Data' table. The table contains the following fields: 'Time on device' (13:43:16), 'Date (YYYY-MM-DD)' (2010-06-16), 'Timezone (+/- hour:minute)' (a dropdown menu showing '+1:00'), 'Daylight Saving Time (DST)' (checkbox), and '12 hour clock' (checkbox). Below the table are 'UNDO' and 'SUBMIT' buttons.

Configuration Data	
Time on device	13:43:16
Date (YYYY-MM-DD)	2010-06-16
Timezone (+/- hour:minute)	+1:00
Daylight Saving Time (DST)	<input type="checkbox"/>
12 hour clock	<input type="checkbox"/>

Figure 1: Example of WBM clock configuration

5. Set the values in the fields "Time on device", "Date" and "Timezone" to the current values and enable the "Daylight Saving Time (DST)" option if necessary.
6. Click on **[SUBMIT]** to apply the changes in your fieldbus node.
7. Restart the fieldbus node to apply the settings of the Web interface.

8.6 Restoring Factory Settings

To restore the factory settings, proceed as follows:

1. Switch off the supply voltage of the fieldbus controller.
2. Connect the communication cable 750-920 to the configuration interface of the fieldbus controller and to a vacant serial port on your computer.
3. Switch on the supply voltage of the fieldbus controller.
4. Start the **WAGO-ETHERNET-Settings** program.
5. In the top menu bar, select **Default** and click **[Yes]** to confirm.

A restart of the fieldbus node is implemented automatically. The start takes place with the default settings.

9 Programming the PFC using WAGO-I/O-PRO

Using IEC 61131-3 programming, the Programmable Fieldbus Controller ETHERNET 750-881 can also utilize the function of a PLC in addition to the functions of a fieldbus coupler. Creation of an application program in line with IEC 61131-3 is performed using the programming tool WAGO-I/O-PRO.

Note



Activate option "CoDeSys" in the web-based Management System for programming!

Pay attention, the IEC 61131-3 programming of the controller via ETHERNET requires that the check box **CoDeSys** be activated at the Website "Port Configuration" (default).

You can, however, also connect the client PC and controller serially for programming using a programming cable.

A description of programming using WAGO-I/O-PRO is not included in this manual. The following sections, on the other hand, contain important information about creating projects in WAGO-I/O-PRO and about special modules that you can use explicitly for programming of the Programmable Fieldbus Controller ETHERNET.

Explanations are also provided as to how the IEC 61131-3 program is transferred and how suitable communication drivers are loaded.

Information



Additional Information:

For a detailed description of using the software, refer to the manual for the "WAGO-I/O-PRO". This manual is located in the Internet under:
www.wago.com → Documentation → WAGO-I/O-SYSTEM 759 → WAGO-I/O-PRO → 759-333

1. Start the programming tool at **Start \ Programs \ WAGO-I/O-PRO**.
2. Under **File / New** create a new project

A dialog window then appears on which you can set the target system for programming.



Figure 51: Dialog window for target system settings

3. Select the WAGO 750-881 ETHERNET Controller 10/100 Mbit/s; digital and analog Signals 750-881 by entering **WAGO_750-881** and then click **OK**.
4. In the dialog window that appears select the program type (AWL, KOP, FUP, AS, ST or CFC).

To ensure that you can access all I/O module data properly in your new project, first compile the I/O module configuration based on the existing fieldbus node hardware and map it in the configuration file "EA-config.xml".

This file defines whether write access is permitted to the modules from the IEC-61131-3 program, from the MODBUS/TCP or from Ethernet/IP.

As described below, this file can be generated via configuration using the WAGO I/O Configurator.

9.1 Configuration using the WAGO-I/O-PRO I/O Configurator

The I/O Configurator is a plug-in incorporated into WAGO-I/O-PRO for assigning addresses to modules at a controller.

1. In the left half of the screen for the WAGO-I/O-PRO interface, select the tab **Resources**.
2. In the tree structure click **Control system configuration**. The I/O Configurator then starts up.
3. Expand the branch **Hardware configuration** in the tree structure with the sub-branch **K Bus**.
4. Right click on **K Bus** or on an **I/O module** to open the menu for adding and attaching I/O modules.
5. By right clicking on the entry **K Bus** and the command **Attach subelement** in the menu, you can select the required I/O module from the I/O module catalog.
(In the new versions of the I/O Configurator open the I/O module catalog by additional clicking on the button **Add**.)
6. Attach it to the end of the **K Bus** structure using **Insert** and then clicking **OK**. In this case, the command "Insert element" is deactivated.
7. To insert an I/O module in front of a selected I/O module in the K Bus structure, right click on an I/O module that has already been selected and then click **Insert element**. In this case, the command "Insert sub-element" is deactivated.

You can also access these commands with the **Insert** menu in the main window menu bar. The dialog window "I/O configuration" for selecting modules is opened both by **Attach sub-element** and by **Insert element**. In this dialog window, you can position all the required modules in your node configuration

8. Position all of the required I/O modules until this arrangement corresponds to the configuration of the physical node.

Complete the tree structure in this process for each module in your hardware that sends or receives data.



Note

The terminal bus structure in the WAGO I/O Configurator must match the physical node structure!

The number of modules that send or receive data must correspond to the existing hardware (except for supply modules, copying modules or end modules, for example).

For the number of input and output bits or bytes of the individual I/O modules please, refer to the corresponding description of the I/O modules.



Information

Additional Information

To obtain further information about an I/O module, either select that module from the catalog, or in the current configuration and then click the button Data Sheet. The module is then shown in a separate window with its associated data sheet. For the current version of the data sheets go to <http://www.wago.com> under Documentation.

9. Click **OK** to accept the node configuration and close the dialog window.

The addresses for the control system configuration are then recalculated and the tree structure for the configuration updated.

If required, you can also modify the authorization privileges for individual I/O modules if they are to be accessed via fieldbus (MODBUS/TCP/IP or Ethernet/IP). Initially, write access from the PLC is defined for each I/O module that is added. Proceed as follows to change this setting:

10. Click on a module in the configuration.
11. In the right dialog window under the tab "Module parameters" define for each module from where access to the module data is to be carried out.

You can choose from the following settings in the column "Value" for this:

- PLC (standard setting) - Access from PLC
- fieldbus 1 - Access from MODBUS/TCP
- fieldbus 2 - Access from Ethernet/IP

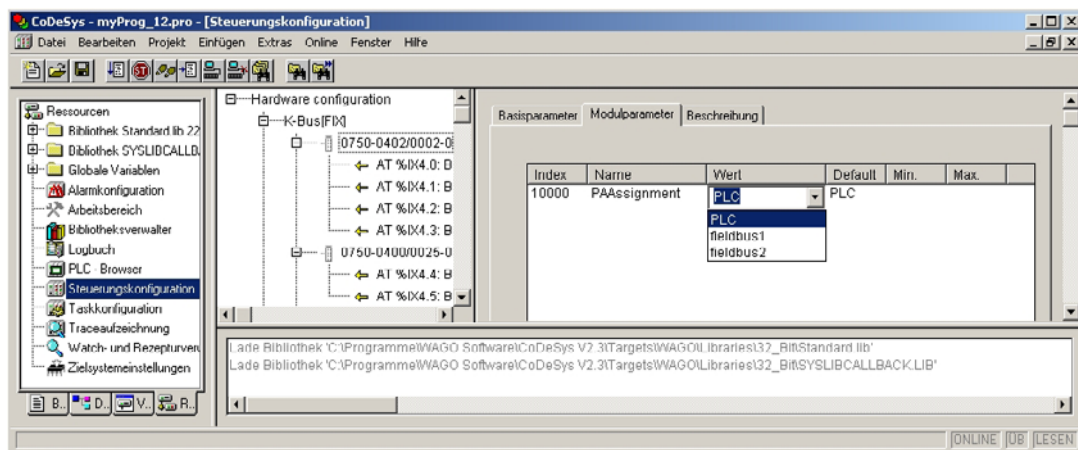


Figure 52: Write access via module parameters

After completing these settings you can begin the IEC-61131-3 programming.

An "EA-config.xml" configuration file is automatically generated and stored in the fieldbus controller, when you transfer the project (Menu **project > transfer/transfer all**) and download it in the fieldbus controller.

Note



Set "fieldbus1", when directly writing to a hardware address via MODBUS!
Set fieldbus 1 if you wish to write directly to a hardware address via MODBUS. Otherwise the modules will be allocated to the PLC, making writing from a different location impossible.

Information



Additional Information

For a detailed description of using the software, refer to the manual for the "WAGO-I/O-PRO". This manual is located in the Internet under: www.wago.com
→ Documentation → WAGO-I/O-SYSTEM759 → WAGO-I/O-PRO → 759-333

9.1.1 Configuration using the "EA-config.xml" File

You can also create the file "EA-config.xml" using an editor and store it in the controller directory "/etc" by means of FTP.

Configuration using the file "EA-config.xml" that is already stored in the controller is described in this section.



Note

Configuration entries in WAGO-I/O-PRO overwrite "EA-config.xml" upon download!

If you wish to perform module assignment directly using the "EAconfig.xml" file stored in the controller, do not save any configuration data in WAGO-I/O-PRO prior to this, as the file is overwritten by entries in the WAGO-I/O-PRO on each download.

1. Open any FTP client. You can also use the Windows FTP client in the DOS prompt window:

`ftp://[IP address of controller]`, e.g. `ftp://192.168.1.201`

2. Then, enter **admin** as the user login and **wago** as the password..

The file "EA-config.xml" is located in the "etc" folder.

3. Copy this file to a local directory on your PC and open it in an editor installed on your PC (e.g., "WordPad").

The file already contains the following syntax:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="\\cplcfg\EA-config.xsl" ?>
<WAGO>
<Module ARTIKELNR="" MAP="PLC" LOC="ALL"></Module>
</WAGO>
```

Figure 53: EA-config.xml

The fourth line contains the necessary information for the first I/O module. The entry MAP=PLC assigns write access privileges to the IEC-61131-3 program for the first module.

4. If you want to enable access via MODBUS/TCP, replace "PLC" with "FB1" and for access from Ethernet/IP, replace "PLC" with "FB2":

```
<Module ARTIKEL NUMBER="" MAP="PLC" LOC="ALL">
↓
<Module ARTIKEL NUMBER="" MAP="FB1" LOC="ALL">
```

```
<Module ARTIKEL NUMBER=" " MAP="FB1" LOC="ALL">  
</Module>
```

5. Then complete the fourth line for each individual module using this syntax and set the corresponding assigned access privileges.

Note



The number of line entries must correspond with the number of bus terminals used!

It is imperative that the number of line entries concurs with the number of existing hardware modules.

6. Save the file and reload it to the controller file system via FTP client.

You can then begin with IEC-61131-3 programming.

Information



Additional Information:

For a detailed description of how to use the software, refer to the WAGO-I/O-PRO manual. The manual is available in the Internet under:

<http://www.wago.com> → Documentation → WAGO Software → WAGO-I/O-PRO → 759-333

9.2 ETHERNET Libraries for WAGO-I/O-PRO

Various libraries are available in WAGO-I/O-PRO for different IEC 61131-3 programming tasks. These libraries contain function blocks that can be used universally to facilitate and accelerate the creation of programs.

Once the libraries have been integrated, function blocks, functions and data types will be available that you can use the same as ones you have specifically defined.



Information

Additional Information

All libraries are included on the installation CD for the software WAGO-I/O-PRO or in the Internet under: <http://www.wago.com> → Downloads → AUTOMATION

The libraries described below are specific to ETHERNET projects with WAGO-I/O-PRO.

Table 37: ETHERNET libraries for WAGO-I/O-PRO

Library	Description
Ethernet.lib	Function blocks for communication via ETHERNET
WAGOLibEthernet_01.lib	Function blocks that can set up a link to a remote server or client via TCP protocol to exchange data with any potential UDP server or client via UDP protocol
WAGOLibModbus_IP_01.lib	Function blocks that set up links with one or more slaves
ModbusEthernet_04.lib	Function blocks for data exchange with several MODBUS/TCP/UDP slaves Also a function block that provides a MODBUS server that maps the MODBUS services on a word array.
SysLibSockets.lib	Function block for access to sockets for communication via TCP/IP and UDP.
WagoLibSockets.lib	Function blocks for access to sockets for communication via TCP/IP and UDP Contains additional functions in addition to SysLibSockets.lib.
Mail_02.lib	Function block for sending e-mails
WAGOLibMail_02.lib	Function block for sending e-mails
WagoLibSnmpEx_01.lib	Function blocks for sending SNMP-V1 traps together with the parameters for the type DWORD and STRING(120) (starting with software version SW >= 07).
WagoLibSntp.lib	Function blocks for setting and using the simple network time protocol (SNTP)
WagoLibFtp.lib	Function blocks for setting and using the file transfer protocol (FTP)
WAGOLibTerminalDiag.lib	Function blocks for the output of module, channel and diagnostic data of I/O modules that provide diagnostic data

Information



Additional Information

For a detailed description of the function blocks and use of the software, refer to the online Help function for *WAGO-I/O-PRO* or the *WAGO-I/O-PRO* manual in the Internet under: <http://www.wago.com> → Documentation → WAGO Software → *WAGO-I/O-PRO* → 759-333.

9.3 Functional Restrictions and Limits

The basis of WAGO-I/O-PRO, the standard programming system CoDeSys by 3S, has an integrated visualization. Dependend on the target, this visualization can be used in the variants "HMI", "TargetVisu" and "WebVisu".

The fieldbus controller supports the process variants "HMI" and "WebVisu". Depending on the version, there are technological limitations.

Several options for complex visualization objects "Alarm" and "Trend" are only provided by the "HMI" version. This applies, for example, to sending emails as a response to an alarm or for navigating through and generating historical trend data.

Compared with "HMI," the "WebVisu" on the fieldbus controller is executed within considerably tighter physical limits. Whereas the "HMI" can call upon the resources of a PC, the "WebVisu" operate within the following restrictions:

File system (2 MB):

The overall size of the PLC program, visualization files, bitmaps, log files, configuration files, etc. must fit into the file system.

The PLC browser delivers the amount of free disk space in response to the command "fds" (FreeDiscSpace).

Process data buffer (16 kB):

The WebVisu uses its own protocol for exchanging process data between applet and control system. In doing so, the process data is transmitted with ASCII coding. The pipe character ("|") separates two process values. For this reason, the required space of a process data variable in the process data buffer not only depends on the data type, but also on the process values itself. A "WORD" variable therefore occupies between one byte for the values 0..9 and five bytes for values greater than 10000. The selected format allows only a rough estimate of the space required for the individual process data in the process data buffer. If the size is exceeded, the WebVisu no longer works as expected.

The number of modules (1023/default):

The total size of the PLC program is determined, among other things, by the maximum number of modules. This value can be configured in the target system settings.

Computing power/processor time:

The 750-881 is based on a real-time operating system with pre-emptive multitasking. High-priority processes such as the PLC program will eliminate low-priority processes.

The web server supplies process data and applets for the web visualization. Make sure when configuring tasks, that there is sufficient processor time available for all processes. The "freewheeling" task call option is not suitable in conjunction with the "WebVisu"; as in this case, the high-priority PLC program suppresses the web server. Instead of this, use the "cyclic" task call option with a realistic value. The PLC browser provides an overview of the real execution times for all

CoDeSys tasks with the command "tsk".

If in a PLC program, operating system functions are used; e.g., for the handling of "sockets" or the "file system," these execution times are not taken into consideration covered by the command "tsk".

CTU counter:

The CTU counter operates in a value range of 0 to 32767.

Network load:

The Programmable Fieldbus Controller ETHERNET has one CPU responsible both for running the PLC program and for handling network traffic.

Ethernet communication demands that every telegram received is processed, regardless of whether it is intended for the Programmable Fieldbus Controller ETHERNET or not.

A significant reduction of the network load can be achieved by configuring the bandwidth limit of the integrated switch module or by using external "switches" instead of "hubs".

However, broadcast telegrams can either only be checked by the sender or with configurable switches that have broadcast limiting. A network protocol analyzer/monitor such as www.ethereal.com provides an overview of current network loading.

Note



Do not use bandwidth limits to increase the operational safety!

The bandwidth limit that can be configured in the WBM under the "Ethernet" link is not suitable for increasing the operating reliability of the "WebVisu", as in this case telegrams are ignored or rejected.

Information



Additional Information

The definition of hard benchmark data is not possible (due to the reasons mentioned above). For planning support, please use the application notes published online for relevant projects featuring the capability of Web visualization. This information is located at <http://www.wago.com>.



Note

Note the maximum number of write cycles of the EEPROM!

Fieldbus couplers/controllers save some information such as IP addresses and IP parameters in the EEPROM to make it available after a restart. The memory cycles of an EEPROM are generally limited. Beyond a limit of approx. 1 million write cycles, memory can no longer be assured. A defective EEPROM only becomes apparent after a restart by software reset or power-on.

Due to a bad checksum, the fieldbus coupler/controller then always starts with the default parameters.

The following functions use the EEPROM:

- **WAGO-I/O-PRO**
 - **WagoLibDaylightSaving** SetDaylightSavings
 - **EthernetLib** SetNetworkConfig
SetVariables
- **MODBUS**
 - Register 0x1035 Time Offset
 - Register 0x100B Watchdog parameters
 - Register 0x1028 Network configuration
 - Register 0x1036 Daylight saving
 - Register 0x1037 Modbus response delay
 - Register 0x2035 PI parameter
 - Register 0x2043 Default configuration
- **Ethernet/IP**
 - Class 0xF5
 - Class 0xF6
 - Class 0x64
- **Parameter assignments**
 - **BootP** new parameters
 - **DHCP** new parameters
 - **WAGO MIB** write access

9.4 General Information about IEC Tasks

Please note the following information when programming your IEC tasks:



Note

Use different priorities for IEC tasks!

IEC tasks must have different priorities, as otherwise an error will occur during translating of the application.

An interruption of IEC tasks is possible through tasks of higher priority!

An ongoing task may be interrupted by tasks with higher priorities. Execution of the task that has been interrupted is resumed only when there are no other higher-priority tasks to be executed.

Distortion of variables in overlapping areas of the process image!

If several IEC tasks utilize input or output variables with the same, or overlapping addresses in the process image, the values for the input or output variables may change while the IEC task is being executed!

Observe waiting periods of free-running tasks!

Running tasks are halted after each task cycle for half the time that the task proper requires (min. 1 ms). Execution of the task is then resumed.

Example: 1st Task 4 ms → Waiting period 2 ms
 2nd Task 2 ms → Waiting period 1 ms

The default task is created by default!

If no task has been defined in the task configuration, a running default task is created during translation. This task, called "Default task," is recognized by this name in the firmware, meaning that the name "Default task" can not be used for other task names.

Observe the watchdog sensitivity for cyclic tasks!

The watchdog sensitivity indicates how many times the watchdog time is exceeded for an even to be triggered. You set the sensitivity in *WAGO-I/O-PRO* under Register **Resources > Task Configuration** for Cyclical Tasks. The values 1 and 0 are equivalent with regard to sensitivity. A sensitivity value of 0 or 1 results in the watchdog event being triggered when the watchdog time is exceeded on time. With a sensitivity value of 2, for instance, the watchdog time must be exceeded in two consecutive task cycles in order for the watchdog event to be triggered.

The following applies to cyclic tasks with watchdog activated:

Note



Reference for Watchdog Settings!

For each task created, a watchdog can be enabled that monitors the execution time of a task.

If the task runtime exceeds the specified watchdog time (e.g., $t_{\#200\text{ ms}}$), then the watchdog event has occurred.

The runtime system stops the IEC program and reports an error.

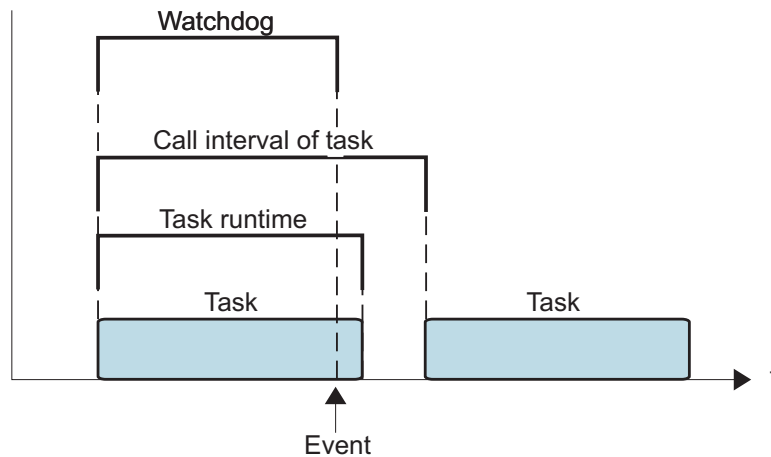


Figure 54: Watchdog runtime is less than the task runtime

If the watchdog time set is greater than the call interval of the task, then the watchdog is restarted for each task call interval.

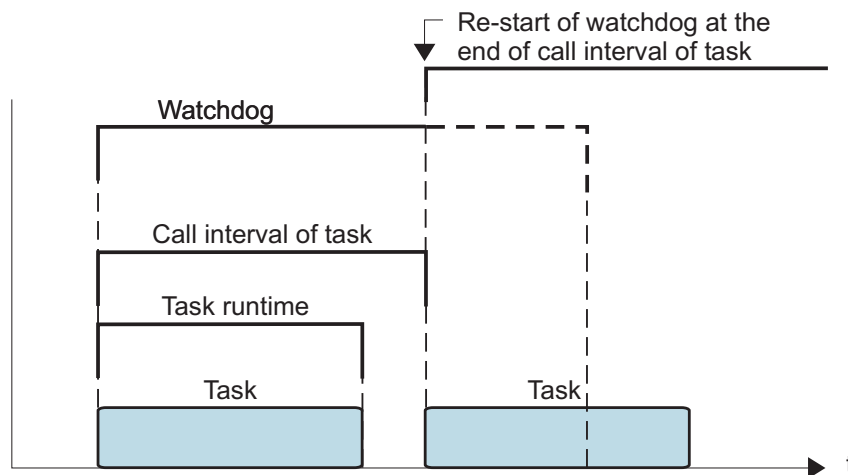


Figure 55: Watchdog runtime is greater than the task call interval

To cyclic tasks applies:



Note

Cyclic tasks with > 30 min. call intervals not possible!

Cyclic tasks with a call interval of more than 30 minutes are not possible.

9.4.1 IEC Task Sequence

1. Determine the system time (tStart).
2. If no full internal bus cycle has run since the last time the outputs were written:
→ Wait until the next internal bus cycle is completed.
3. Reading of inputs and reading back of the outputs from the process image.
4. If the application program has been started.
→ Execute the program codes for this task.
5. Writing of the outputs to the process image.
6. Determine the system time (tEnd).
→ $tEnd - tStart = \text{runtime for the IEC task}$

9.4.2 Overview of Most Important Task Priorities

Table 38: Task processing

Task	Importance of the execution
Internal bus task, fieldbus task	of priority before all others
Normal task	after the internal bus and fieldbus tasks
PLC-Comm task	after the normal tasks
Background task	after the PLC-Comm tasks

I/O Bus Task / Fieldbus Task (Internal)

The I/O Bus task is an internal task, which updates the I/O module data from the process image. Fieldbus tasks are triggered by fieldbus events (communications); therefore, they only use processing time when the fieldbus is active (MODBUS/Ethernet/IP).

Normal task (IEC tasks 1-10)

IEC tasks with this priority may be interrupted by the internal bus tasks. Therefore, configuration for the connected modules and communication via fieldbus with the watchdog activated for the task call interval must be taken into account here.

PLC-Comm task (internal)

The PLC-Comm task is active when logged in and takes up communication with the CoDeSys gateway.

Background task (IEC-Task priorities 11-31 that can be set in CoDeSys)

All internal tasks have a priority higher than that for the IEC background tasks. These tasks are therefore very well-suited for performing time-intensive and non-critical time tasks, such as calling up functions in the SysLibFile.lib.

Information

**Additional Information**

For a detailed description of using the software, refer to the manual for the "WAGO-I/O-PRO". This manual is located in the Internet under:

www.wago.com → Documentation → WAGO Software → WAGO-I/O-PRO → 759-333.

9.5 System Events

In place of a task, a system event can also call up a project module for processing.

The system events to be employed for this depend on the target system. These events consist of the list of supported standard system events for the control system and any other manufacturer-specific events which may have been added.

Possible events, for example: Stop, Start, Online change.

A complete list of all system events is provided at *WAGO-I/O-PRO* in tab **Resources > Task configuration > System events**.

9.5.1 Enabling/disabling system events

1. Open the register **resources > task configuration > system events** in *WAGO-I/O-PRO* (see the following Figure).
2. In order to call up a module via an event, activate the entries by setting a hatch mark in the respective control boxes.
3. Disable the control boxes by removing the hatch marks through a mouse click.

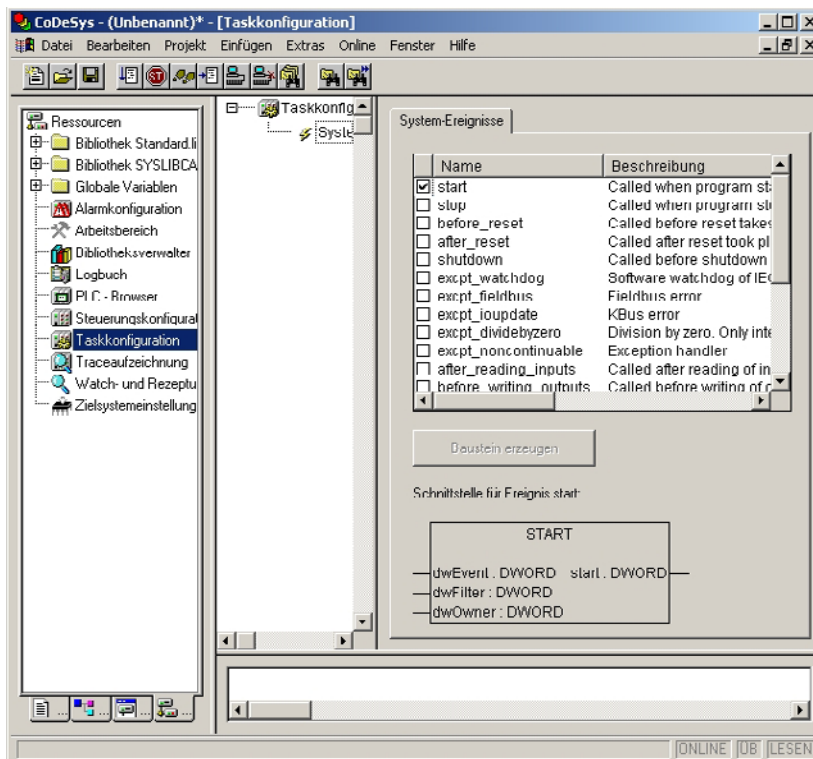


Figure 56: System events

Information

**Additional Information:**

Allocation of the system events to the specific modules to be called up is clarified in the manual for the programming tool WAGO-I/O-PRO in the Internet under:
<http://www.wago.com> → Documentation → WAGO Software → WAGO-I/O-PRO → 759-333

9.6 Transfer the IEC program to the controller

Transfer from the PC to the controller of the program for the created IEC-61131-3 application can be performed two ways (see following sections).

- Direct transfer via serial RS-232 port
- Transfer by means of TCP/IP via fieldbus

Suitable communication drivers are required for transfer; these can be loaded and configured using *WAGO-I/O-PRO*.

Note



Check/adjust communications parameters of the driver

When selecting the desired driver, watch for the proper settings and adjustments of the communications parameters (see the following description).

Note



"Reset" and "Start" are required to set the physical outputs!

The initialization values for the physical outputs are not set immediately after downloading. Select **Online** > **Reset** and subsequently **Online** > **Start** in the menu bar of *WAGO I/O-PRO* to set the values.

Note



Stop application before generating large boot projects!

Stop the *WAGO-I/O-PRO* application via **Online** > **Stop** before generating a very large boot project, since this may otherwise cause stopping the internal bus. You can restart the application after creating the boot project.

Note



Handling persistent data affects the program start!

Depending on the variable type, the number and sizes of the persistent data and their combination, such as in function modules, handling with persistent data can delay the program start by an extended initialization phase.

Information



Additional Information

The following description is used for fast access. For details on installing missing communication drivers and using the software, refer to "*WAGO-I/O-PRO*" available in the Internet under:

www.wago.com → Service → Downloads → Documentation → WAGO Software → *WAGO-I/O-PRO*.

9.6.1 Transfer via Serial Service Port



Note

Watch the position of the mode selector switch when accessing the controller!

Prerequisite for the access to the fieldbus controller is that the operating mode switch of the controller, which is located behind the cover of the fieldbus controller next to the service interface, is in the center or top position.

Use the WAGO communication cable to set up a physical connection via serial service port. This cable is included in the scope of supply for the IEC-61131-3 programming tool (order no. 759-333), or can be procured as an accessory item under order no. 750-920.

NOTICE

Do not connect 750-920 Communication Cable when energized!

To prevent damage to the communications interface, do not connect or disconnect 750-920 Communication Cable when energized! The fieldbus controller must be de-energized!

1. Check that the controller mode selector switch is set to the center or top position.
If this is not the case, move the mode selector switch to the center or top position.
2. Use the WAGO communication cable to connect a COM port of your PC to the controller communication port.

A communication driver is required for serial data transfer. This driver and its parameters must be entered in the WAGO-I/O-PRO in the dialog window "Communication parameters".

3. Start the WAGO-I/O-PRO software **under Start > Programs > WAGO Software > WAGO-I/O-PRO.**
4. In the menu **Online** select the item **Communication parameters.**

The dialog window "Communication parameters" then appears. The channels of the currently connected gateway servers are shown on the left side of the dialogue and the already installed communications drivers are shown below. This window is empty in its default settings.

5. Click **New** to set up a link and then enter a name, such as RS-232 Connection.

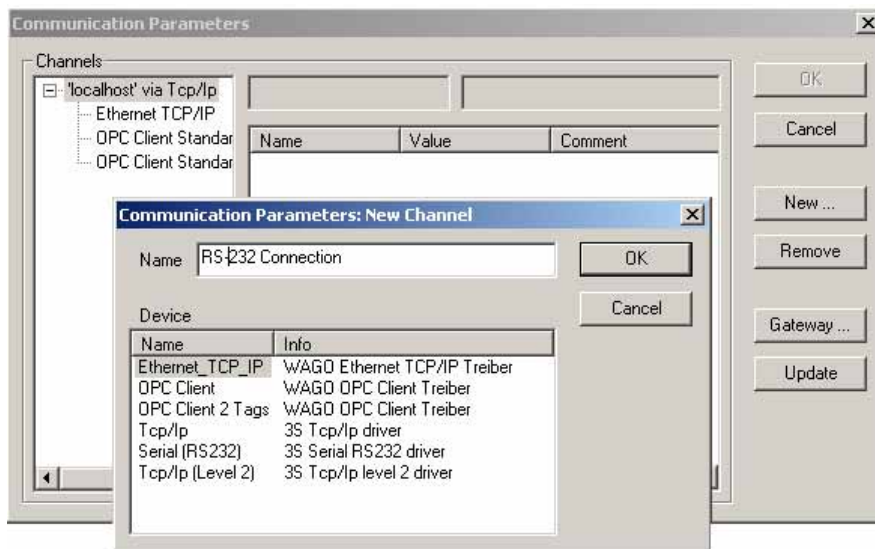


Figure 57: Dialog window "Communication parameters"

- In the selection window, mark the required driver in the right side of the window, Serial (RS-232) 3S Serial RS-232 driver, to configure the serial link between the PC and the controller.

The following properties for the serial port are shown in the center dialog window:

- Port: COM1
- Baud rate: 19200
- Parity: Even
- Stop-bits: 1
- Motorola byte order: No

- If necessary, change the entries according to the above values by clicking on the respective value and editing it.
- Confirm these settings by clicking **OK**

The RS-232 port is now configured for transferring the application.

- Under **Online**, click the menu item **Login** to log in to the controller

The WAGO-I/O-PRO Server is active during online operation. The communication parameters can not be called up during this time.

Depending on whether a program is already present in the controller, a window will appear asking whether a (new) program should be loaded.

- Respond with **Yes** to load the current program.
- In menu **Online**, click on **Create Boot project**.

You compiled project will also be executed by this method, if you restart the controller or if there is a power failure.

12. Once the program has been loaded, start program processing in the menu **Online**, menu item **Start**.

This command starts the processing of your program in the control system or in the simulation.

"ONLINE" and "RUNNING" will then appear at the right of the status bar.

13. To terminate online operation, click the menu item **Log off** in the menu **Online**.

9.6.2 Transfer via Fieldbus and ETHERNET

The physical link between the PC and the controller is set up via fieldbus. An appropriate communication driver is required for data transfer. The driver and its parameters must be entered in the WAGO-I/O-PRO in the dialog window "Communication parameters".



Note

Controller needs IP address for access!

The controller must have an IP address before it can be accessed. The operating mode switch, which is located behind the cover of the fieldbus controller next to the service interface, must be in the center or top position.

1. Start the WAGO-I/O-PRO software under **Start / Programs / WAGO-I/O-PRO** or by clicking the program icon on the desktop).
2. In the menu **Online** select the item **Communication parameters**.

The dialog window "Communication parameters" then appears. The channels of the currently connected gateway servers are shown on the left side of the dialogue and the already installed communications drivers are shown below. This window is empty in its default settings.

3. Click **New** to set up a connection and then specify a name, e.g. TcpIp connection.
4. Mark the required TCP/IP driver in the right side of the dialog window to configure the link between the PC and the controller via ETHERNET. Use the new driver version "Tcp/Ip" (3S Tcp/Ip driver).

The following standard entries are shown in the center dialog window:

- IP address: IP address of your controller
- Port number: 2455
- Motorolabyteorder: No
- Debug level: 16#0000

5. Change any entries as you may require.
6. Confirm with **OK**.

You have now configured the TCP/IP link with the communication parameters/drivers.

7. Under **Online**, click the menu item **Login** to log in to the controller

The WAGO-I/O-PRO Server is active during online operation. The communication parameters can not be called up during this time.

Depending on whether a program is already present in the controller, a window will appear asking whether a (new) program should be loaded.

8. Respond with **Yes** to load the current program.
9. In menu **Online**, click on **Create Boot project**.

Your compiled project will also be executed by this method, if you restart the controller or if there is a power failure.

10. Once the program has been loaded, start program processing in the menu **Online**, menu item **Start**.

This command starts the processing of your program in the control system or in the simulation.

"ONLINE" and "RUNNING" will then appear at the right of the status bar.

11. To terminate online operation, click the menu item **Log off** in the menu **Online**.

10 Configuring via the Web-Based Management System (WBM)

An internal file system and an integrated Web server can be used for configuration and administration of the system. Together, they are referred to as the Web-Based Management System (WBM).

The HTML pages saved internally provide you with information about the configuration and status of the fieldbus node. In addition, you can also change the configuration of the device here.

You can also save HTML pages created yourself via the implemented file system.

Note



Always restart after making changes to the configuration!

The system must always be restarted for the changed configuration settings to take effect.

1. To open the WBM, launch a Web browser (e.g., Microsoft Internet Explorer or Mozilla Firefox).
2. Enter the IP address of the fieldbus coupler/controller in the address bar (192.168.1.1 by default or as previously configured).
3. Click **[Enter]** to confirm.
The start page of WBM loads.
4. Select the link to the desired HTML page in the left navigation bar.
A query dialog appears.
5. Enter your user name and password in the query dialog (default: user = "admin", password = "wago" or user = "user", password = "user").
The corresponding HTML page is loaded.
6. Make the desired settings.
7. Press **[SUBMIT]** to confirm your changes or press **[UNDO]** to discard the changes.
8. Restart the system to apply the settings.

10.1 Information

The WBM page "Information" contains an overview of all important information about your fieldbus coupler/controller.

WAGO INNOVATIVE CONNECTIONS

Web-based Management

WAGO Kontakttechnik GmbH & Co. KG
Hansstr. 27
D-32423 Minden
www.wago.com

Navigation

- Information
- Ethernet
- TCP/IP
- Port
- SNMP
- SNMP V3
- Watchdog
- Clock
- Security
- PLC
- Features
- IO config
- WebVisu

Status information

Coupler details

Order number	750-881
Mac address	0030DE038178
Firmware revision	01.01.13 (01)

Actual network settings

IP address	192.168.1.206
	Static Configuration
Subnet mask	255.255.0.0
Gateway	0.0.0.0
Hostname	
Domainname	
(S)NTP-Server	0.0.0.0
DNS-Server 1	0.0.0.0
DNS-Server 2	0.0.0.0

Module status

State Modbus Watchdog:	Disabled
Error code:	0
Error argument:	0
Error description:	Coupler running, OK

Figure 58: WBM page "Information"

Table 39: WBM page "Information"

Coupler details			
Entry	Default	Value (example)	Description
Order number	750-881/000-000	750-881/000-000	Order number
Mac address	0030DEXXXXXX	0030DE000006	Hardware MAC address
Firmware revision	kk.ff.bb (rr)	01.01.09 (00)	Firmware revision number (kk = compatibility, ff = functionality, bb = bugfix, rr = revision)
Actual network settings			
Entry	Default	Value (example)	Description
IP address	192.168.1.1	192.168.1.80	IP address, Type of IP address assignment
Subnet mask	255.255.255.0	255.255.255.240	Subnet mask
Gateway	0.0.0.0	192.168.1.251	Gateway
Hostname			Host name (not assigned here)
Domainname			Domain name (not assigned here)
(S)NTP server	0.0.0.0	0.0.0.0	Address of (S)NTP server
DNS server 1	0.0.0.0	0.0.0.0	Address of first DNS server
DNS server 2	0.0.0.0	0.0.0.0	Address of second DNS server
Module status			
Entry	Default	Value (example)	Description
State Modbus Watchdog	Disabled	Disabled	Status of Modbus Watchdog
Error code	0	10	Error code
Error argument	0	5	Error argument
Error description	Coupler running, OK	Mismatch in CoDeSys IO-configuration	Error description

10.2 Ethernet

Use the "Ethernet" HTML page to set the data transfer rate and bandwidth limit for each of the two switch ports for data transfer via Ethernet.

Navigation

- Information
- Ethernet
- TCP/IP
- Port
- SNMP
- SNMP V3
- Watchdog
- Clock
- Security
- PLC Info
- PLC
- Features
- IO config
- Disk info
- WebVisu

Ethernet configuration

This page is for the configuration of the Ethernet Switch and Aging settings. The configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

Phy Configuration

Desc	Port 1	Port 2
Enable Port	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Enable Autonegotiation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10 Mbit Half Duplex	<input type="checkbox"/>	<input type="checkbox"/>
10 Mbit Full Duplex	<input type="checkbox"/>	<input type="checkbox"/>
100 Mbit Half Duplex	<input type="checkbox"/>	<input type="checkbox"/>
100 Mbit Full Duplex	<input type="checkbox"/>	<input type="checkbox"/>

Undo Submit

Chip Configuration

Description	Enable?
Fast Aging	<input type="checkbox"/>

Misc Configuration

Desc	Port 1	Port 2	Internal Port
Input Limit Rate	[4Kbps]	[4Kbps]	[4Kbps]
Output Limit Rate	[No Limit]	[No Limit]	[No Limit]
BC protection	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Port Mirror	<input type="checkbox"/>		
Sniffer Port	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Minor Port	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ethernet MTU	[1500]		

Undo Submit

Figure 59: WBM-Seite „Ethernet“

Table 40: WBM page "Ethernet"

Phy Configuration				
Entry	Default		Description	
Enable Port	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> Enable Port 1/Port 2	
			<input type="checkbox"/> Disable Port 1/Port 2	
Enable autonegotiation	<input checked="" type="radio"/>		<input checked="" type="radio"/> Enable Autonegotiation Automatically set the best possible transmission speed with "Enable Autonegotiation".	
			<input type="radio"/> Enable Autonegotiation	
10 MBit Half Duplex	<input type="radio"/>		Select half or full duplex for the ETHERNET to configure a fixed transmission speed 10 or 100 MBit	
10 MBit Full Duplex	<input type="radio"/>			
100 MBit Half Duplex	<input type="radio"/>			
100 MBit Full Duplex	<input type="radio"/>			
Chip Configurtaion				
Entry	Default value		Description	
Fast Aging	<input type="checkbox"/>		<input checked="" type="checkbox"/> Enable "Fast Aging" "Fast Aging" ensures that the cache for the MAC addresses is cleared faster in the switch. This may be required if a redundancy system (e.g., using a Jet-Ring network or comparable technology) needs to be set up.	
			<input type="checkbox"/> Disable "Fast Aging"	
Misc. Configuration				
Entry	Port			Description
	1	2	in- ternal	
Input Limit Rate	No Limit ▼			The Input Limit Rate limits network traffic when receiving. The rate is indicated in megabytes or kilobytes per second. If the limit is exceeded, packets are lost.
Output Limit Rate	No Limit ▼			The Output Limit Rate limits network traffic when sending. The rate is indicated in megabytes or kilobytes per second. If the limit is exceeded, packets are lost.
BC protection	<input type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/> Broadcast Protection limits the number of broadcast telegrams per unit of time. If protection is on, the broadcast packets are limited at 100 Mbit to 8 packets per 10 ms and at 10 Mbit to 8 packets per 100 ms. If the limit is exceeded, packets are lost.
				<input type="checkbox"/> Broadcast Protection disabled.
Port Mirror	<input type="checkbox"/>			<input checked="" type="checkbox"/> Enable port mirroring Port Mirroring is used for network diagnostics. Packets are mirrored from one port (mirror port) to another (sniffer port).
				<input type="checkbox"/> Disable port mirroring
Sniffer Port	<input type="radio"/>	<input checked="" type="radio"/>		Select the sniffer port the mirror port should be mirrored to.
Mirror Port	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Select the mirror port which should be mirrored to the sniffer port.
Ethernet MTU	1500			Maximum packet size of a protocol, which can be transferred without fragmentation ("Maximum Transmission Unit" - MTU)

Note**Set the MTU value for fragmentation only!**

Only set the value for MTU, i.e., the maximum packet size between client and server, if you are using a tunnel protocol (e.g., VPN) for ETHERNET communication and the packets must be fragmented.

Setting the value is independent of the transmission mode selected.

Note**Configure ETHERNET transmission mode correctly!**

A fault configuration of the ETHERNET transmission mode may result in a lost connection, poor network performance or faulty performance of the fieldbus coupler/controller.

Note**All ETHERNET ports cannot be disabled!**

Both ETHERNET ports can be switched off. If both ports are disabled and you press [SUBMIT], the selection is not applied and the previous values are restored.

10.3 TCP/IP

You can configure network addressing and network identification on the "TCP/IP" HTML page.


Note



Set the DIP switch to "0" and enable "use IP from EEPROM"!

Before you change parameters on this page, set the DIP switch to zero and on the "Port configuration" WBM page, set the "use IP from EEPROM" option!

If these conditions are not met, the DIP switch settings are applied instead.



WAGO
INNOVATIVE CONNECTIONS

Web-based Management

WAGO Kontakttechnik
GmbH & Co. KG
Hansstr. 27
D-32423 Minden
www.wago.com

Navigation

- [Information](#)
- [Ethernet](#)
- [TCP/IP](#)
- [Port](#)
- [SNMP](#)
- [SNMP V3](#)
- [Watchdog](#)
- [Clock](#)
- [Security](#)
- [PLC](#)
- [Features](#)
- [IO config](#)
- [WebVisu](#)

TCP/IP configuration

This page is for the configuration of the basic TCP/IP network parameters. The parameters are stored in an EEPROM and changes will take effect after the next software or hardware reset.

Note that these settings are used only if the DIP switch is set to zero and you have selected 'use IP from EEPROM' at 'Port' configuration page! Otherwise the settings from DIP switch will be used!

EEPROM Configuration Data

IP-Address	192.168.1.206
Subnet Mask	255.255.0.0
Gateway	0.0.0.0
Hostname	
Domain name	
DNS-Server1	0.0.0.0
DNS-Server2	0.0.0.0
Switch IP-Address	192.168.1
(S)NTP-Server	0.0.0.0
SNTP Update Time (sec, max. 65535)	3600

UNDO
SUBMIT

Figure 60: WBM page "TCP/IP"

Table 41: WBM page „TCP/IP“

Configuration Data			
Entry	Default	Value (example)	Description
IP address	192.168.1.0	192.168.1.200	Enter IP address
Subnet mask	255.255.255.0	255.255.255.0	Enter subnet mask
Gateway	0.0.0.0	0.0.0.0	Enter gateway
Host name			Enter host name
Domain name			Enter domain name
DNS Server1	0.0.0.0	0.0.0.0	Enter IP address of the first DNS server
DNS Server2	0.0.0.0	0.0.0.0	Enter optional IP address of the second DNS server
Switch IP-Address	192.168.1	192.168.5	Network address for the configuration of the IP address with DIP switch
(S)NTP Server	0.0.0.0	0.0.0.0	Enter IP address of the (S)NTP server
SNTP Update Time (sec. max. 65535)	0	0	Enter the delay after which the (S)NTP server requests the network time again

10.4 Port

Use the "Port" HTML page to enable or disable services available via the IP protocol.

WAGO Web-based Management

WAGO Kontakttechnik GmbH & Co. KG
Hansstr. 27
D-32423 Minden
www.wago.com

Navigation

- Information
- Ethernet
- TCP/IP
- Port**
- SNMP
- SNMP V3
- Watchdog
- Clock
- Security
- PLC
- Features
- IO config
- WebVisu

Port configuration

This page is for the configuration of the network protocols. The configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

Port Settings

Protocol	Port	Enabled
FTP	21	<input checked="" type="checkbox"/>
SNTP	123	<input type="checkbox"/>
HTTP	80	<input checked="" type="checkbox"/>
SNMP	161, 162	<input type="checkbox"/>
Ethernet IP	44818 (TCP), 2222 (UDP)	<input checked="" type="checkbox"/>
Modbus UDP	502	<input checked="" type="checkbox"/>
Modbus TCP	502	<input checked="" type="checkbox"/>
WAGO Services	6626	<input checked="" type="checkbox"/>
CoDeSys	2455	<input checked="" type="checkbox"/>
BootP	68	<input type="radio"/>
DHCP	68	<input type="radio"/>
use IP from EEPROM	--	<input checked="" type="radio"/>

UNDO SUBMIT

Figure 61: WBM page "Port"

Table 42: WBM page "Port"

Port Settings		
Entry	Entry	Entry
FTP (Port 21)	Enabled <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> activating "File Transfer Protocol"
		<input type="checkbox"/> deactivating "File Transfer Protocol"
SNTP (Port 123)	Enabled <input type="checkbox"/>	<input checked="" type="checkbox"/> activating "Simple Network Time Protocol"
		<input type="checkbox"/> deactivating "Simple Network Time Protocol"
HTTP (Port 80)	Enabled <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> activating "Hypertext Transfer Protocol"
		<input type="checkbox"/> deactivating "Hypertext Transfer Protocol"
SNMP (Port 161, 162)	Enabled <input type="checkbox"/>	<input checked="" type="checkbox"/> activating "Simple Network Management Protocol"
		<input type="checkbox"/> deactivating "Simple Network Management Protocol"
Ethernet IP (TCP-Port 44818, UDP-Port 2222)	Enabled <input type="checkbox"/>	<input checked="" type="checkbox"/> activating ETHERNET/IP protocol
		<input type="checkbox"/> deactivating ETHERNET/IP protocol
Modbus UDP (Port 502)	Enabled <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> activating MODBUS/UDP protocol
		<input type="checkbox"/> deactivating MODBUS/UDP protocol
Modbus TCP (Port 502)	Enabled <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> activating MODBUS/TCP protocol
		<input type="checkbox"/> deactivating MODBUS/TCP protocol
WAGO Services (Port 6626)	Enabled <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> activating WAGO services
		<input type="checkbox"/> deactivating WAGO services
CoDeSys (Port 2455)	Enabled <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> activating WAGO-I/O-PRO
		<input type="checkbox"/> deactivating WAGO-I/O-PRO
BootP (Port 68)	Enabled <input checked="" type="radio"/>	<input checked="" type="radio"/> activating "Boots Trap Protocol"
		<input type="radio"/> deactivating "Boots Trap Protocol"
DHCP (Port 68)	Enabled <input type="radio"/>	<input checked="" type="radio"/> activating "Dynamic Host Configuration Protocol"
		<input type="radio"/> deactivating "Dynamic Host Configuration Protocol"
use IP from EEPROM	Enabled <input type="radio"/>	<input checked="" type="radio"/> activating use of IP address from EEPROM
		<input type="radio"/> deactivating use of IP address from EEPROM

Note



Alternative IP address assignment!

You can only select the DHCP, BootP and "use IP from EEPROM" settings as an alternative!

10.5 SNMP

On the HTML page "SNMP", you can perform the settings for the Simple Network Management Protocol.

SNMP is a standard for device management within a TCP/IP network. The Simple Network Management Protocol (SNMP) is responsible for transporting the control data that allows the exchange of management information, the status and statistic data between individual network components and a management system.

The fieldbus coupler/controller supports SNMP in versions 1, 2c and 3.

The SNMP of the ETHERNET TCP/IP controller includes the general MIB according to RFC1213 (MIB II).

SNMP is processed via port 161. The port number for SNMP traps (agent messages) is 162.

Note



Enable port 161 and 162 to use SNMP!

Enable ports 161 and 162 in the WBM in menu "port", so that the fieldbus coupler/controller can be reached via SNMP. The port numbers cannot be modified.

Note



Modify parameter via WBM or SNMP objects!

However, parameters that can be set on the html pages can also be changed directly by the appropriate SNMP objects.

Information




Additional Information:

Additional information for SNMP, the Management Information Base (MIB) and traps (event messages via SNMP) may be obtained from chapter "Fieldbus communications" > "Communications protocols" > "SNMP (Simple Network Management Protocol)".

Note that the settings for SNMPV1/V2c and SNMPV3 are separate from each other: The different SNMP versions can be activated or used in parallel or individually on a fieldbus controller.

10.5.1 SNMP V1/V2c

The SNMP version 1/2c represents a community message exchange. The community name of the network community must thereby be specified.



WAGO
INNOVATIVE CONNECTIONS

Web-based Management

WAGO Kontakttechnik
GmbH & Co. KG
Hansstr. 27
D-32423 Minden
www.wago.com

Navigation

- [Information](#)
- [Ethernet](#)
- [TCP/IP](#)
- [Port](#)
- [SNMP](#)
- [SNMP V3](#)
- [Watchdog](#)
- [Clock](#)
- [Security](#)
- [PLC](#)
- [Features](#)
- [IO config](#)
- [WebVisu](#)

SNMP Configuration

This page is dedicated to the SNMP configuration. The new configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

SNMP Configuration

Name of device	<input type="text" value="750-881"/>
Description	<input type="text" value="WAGO Ethernet 750-881"/>
Physical location	<input type="text" value="LOCAL"/>
Contact	<input type="text" value="support@wago.com"/>

SNMP v1/v2c Manager Configuration

Protocol Enable	SNMP V1/V2c <input checked="" type="checkbox"/>
Local Community Name	<input type="text" value="public"/>

SNMP v1/v2c Trap Receiver Configuration

Trap Receiver 1	<input type="text" value="0.0.0.0"/>
Community Name 1	<input type="text" value="public"/>
Trap Version	V1 <input checked="" type="radio"/> V2 <input type="radio"/>
Trap Receiver 2	<input type="text" value="0.0.0.0"/>
Community Name 2	<input type="text" value="public"/>
Trap Version	V1 <input checked="" type="radio"/> V2 <input type="radio"/>

Figure 62: WBM page "SNMP"

Table 43: WBM page "SNMP"

SNMP Configuration		
Entry	Value (Default)	Description
Name of device	750-881	Device name (sysName)
Description	Programmable Fieldbus Controller ETHERNET 750- 881	Device description (sysDescription)
Physical location	LOCAL	Location of device (sysLocation)
Contact	support@wago.com	E-mail contact address (sysContact)
SNMP v1/v2 Manager Configuration		
Entry	Value (Default)	Description
Protocol Enable	SNMP V1/V2c <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activating SNMP Version 1/2c
		<input type="checkbox"/> Deactivating SNMP-Version 1/2c
Local Community Name	public	Used community name
SNMP v1/v2 Trap Receiver Configuration		
Entry	Value (Default)	Description
Trap Receiver 1	0.0.0.0	IP address of 1. used SNMP manager
Community Name 1	public	1. Community name of the network community used
Trap Version	V1 <input checked="" type="radio"/>	V1 <input checked="" type="radio"/> V2 <input type="radio"/> Activating Traps Version 1
	V2 <input type="radio"/>	V1 <input type="radio"/> V2 <input checked="" type="radio"/> Activating Traps Version 2
Trap Receiver 2	0.0.0.0	IP address of 2. used SNMP manager
Community Name 2	public	2. Community name of the network community used
Trap Version	V1 <input checked="" type="radio"/>	V1 <input checked="" type="radio"/> V2 <input type="radio"/> Activating Traps Version 1
	V2 <input type="radio"/>	V1 <input type="radio"/> V2 <input checked="" type="radio"/> Activating Traps Version 2

10.5.2 SNMP V3

In SNMP version 3, exchanging messages is user-related. Each device, that knows the passwords set via WBM, may read or write values from the controller. In SNMPv3, user data from SNMP messages can also be transmitted in encoded form. This is why SNMPv3 is often used in safetyrelated networks.

WAGO INNOVATIVE CONNECTIONS

Web-based Management

WAGO Kontakttechnik
GmbH & Co. KG
Hansastr. 27
D-32423 Minden
www.wago.com

Navigation

- Information
- Ethernet
- TCP/IP
- Port
- SNMP
- SNMP V3**
- Watchdog
- Clock
- Security
- PLC
- Features
- IO config
- WebVisu

SNMP Configuration

This page is dedicated to the SNMP configuration. The new configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.
'Authentication Key' and 'Privacy Key' have to be at least 8 characters.

SNMP v3 (user based)

1.User	activate <input type="checkbox"/>
Authentication Type	None <input type="radio"/> MD5 <input checked="" type="radio"/> SHA1 <input type="radio"/>
Security Authentication Name	<input type="text" value="SecurityName"/>
Authentication Key	<input type="text" value="AuthenticationKey"/>
Privacy Enable	DES <input checked="" type="checkbox"/>
Privacy Key	<input type="text" value="PrivacyKey"/>
Notification/Trap enable	V3 <input type="checkbox"/>
Notification Receiver IP	<input type="text" value="0.0.0.0"/>
2.User	activate <input type="checkbox"/>
Authentication Type	None <input type="radio"/> MD5 <input checked="" type="radio"/> SHA1 <input type="radio"/>
Security Authentication Name	<input type="text" value="SecurityName"/>
Authentication Key	<input type="text" value="AuthenticationKey"/>
Privacy Enable	DES <input checked="" type="checkbox"/>
Privacy Key	<input type="text" value="PrivacyKey"/>
Notification/Trap enable	V3 <input type="checkbox"/>
Notification Receiver IP	<input type="text" value="0.0.0.0"/>

UNDO SUBMIT

Figure 63: WBM page "SNMP V3"

SNMP v3 (user based)		
Entry	Value (Example)	Description
1. User / 2. User	activate <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activating user 1 or 2
		<input type="checkbox"/> Deactivating user 1 or 2
Authentication Type	None <input type="radio"/> MD5 <input checked="" type="radio"/> SHA1 <input type="radio"/>	None <input checked="" type="radio"/> MD5 <input type="radio"/> SHA1 <input type="radio"/> No encryption of the authentication
		None <input type="radio"/> MD5 <input checked="" type="radio"/> SHA1 <input type="radio"/> Encryption of the authentication with MD5
		None <input type="radio"/> MD5 <input type="radio"/> SHA1 <input checked="" type="radio"/> Encryption of the authentication with SHA1
Security Authentication Name	Security Name	Enter the name, if the "authentication type" MD5 or SHA1 has been selected
Authentication Key	Authentication Key	Enter the password with at least 8 characters, if "authentication type" MD5 or SHA1 has been selected
Privacy Enable	DES <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate the DES encryption of the data
		<input type="checkbox"/> Deactivate the DES encryption of the data
Privacy Key	Privacy Key	Enter the password of at least 8 characters in the encryption with DES
Notification/Trap enable	V3 <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate the notification traps of the SNMP version 3
		<input type="checkbox"/> Deactivate the notification traps of the SNMP version 3
Notification Receiver IP	192.168.1.10	IP address of the notification manager

Two independent SNMPv3 users can be defined and activated via the html page (user 1 and user 2).

10.6 Watchdog

Click the link "Watchdog" to go to a Web site where you can specify the settings for the connection and MODBUS watchdog.

WAGO INNOVATIVE CONNECTIONS

Web-based Management

WAGO Kontakttechnik GmbH & Co. KG
Hansstr. 27
D-32423 Minden
www.wago.com

Navigation

- Information
- Ethernet
- TCP/IP
- Port
- SNMP
- SNMP V3
- Watchdog**
- Clock
- Security
- PLC
- Features
- IO config
- WebVisu

Watchdogs

This page is for the configuration of the watchdogs. The configuration is stored in an EEPROM. Changes of the Connection Time will take effect immediately. Changes of the Modbus Watchdog will take effect after the next software or hardware reset. For more information see the manual.

Connection Watchdog

Connection Timeout Value (100ms):

Modbus Watchdog

State Modbus Watchdog:	Disabled
Watchdog Type :	Standard <input checked="" type="radio"/>
	Alternative <input type="radio"/>
Watchdog Timeout Value (100ms):	<input type="text" value="100"/>
Watchdog Trigger Mask (F1 to F16):	<input type="text" value="0xFFFF"/>
Watchdog Trigger Mask (F17 to F32):	<input type="text" value="0xFFFF"/>

Figure 64: WBM page "Watchdog"

Table 44: WBM page "Watchdog"

Connection watchdog		
Entry	Default	Description
Connection Timeout Value (100 ms)	600	Monitoring period for TCP links. After the completion of this period without any subsequent data traffic, the TCP connection is closed.
Modbus Watchdog		
Entry	Default	Description
State Modbus Watchdog	Disabled	Enabled – Watchdog is activated Disabled – Watchdog is disabled
Watchdog Type	Standard <input checked="" type="radio"/>	The set coding mask (watchdog trigger mask) is evaluated to determine whether the watchdog time is reset.
	Alternative <input type="radio"/>	The watchdog time is reset by any Modbus/TCP telegram.
Watchdog Timeout Value (100 ms)	100	Monitoring period for Modbus links. After the completion of this period without receiving a Modbus telegram, the physical outputs are set to "0".
Watchdog Trigger Mask (F 1 to F16)	0xFFFF	Coding mask for certain Modbus telegrams (Function Code FC1 ... FC16)
Watchdog Trigger Mask (F17 to F32)	0xFFFF	Coding mask for certain Modbus telegrams (Function Code FC17 ... FC32)

10.7 Clock

Specify the settings for the internal real-time clock on the "Clock" HTML page. Here, enter the current time and date and also select standard or daylight saving time.

Note

**Reset the internal clock after 6 days without power supply!**

The internal clock must be reset on initial startup or after 6 days without power. If the clock is not set, the clock begins with the date 01.01.2000 around 0:00 clock with time measurement.

Note

**Integrate the function block for converting from winter/summer time!**

Switch-over between standard and daylight saving time via Web-based management system is required when synchronizing the controllers in your network using a time server. The controller itself does not automatically execute a change-over between standard and daylight-saving time. The change-over is resolved via function block PrgDaylightSaving, which you must integrate into the WAGO-I/O-PRO using the library DaylightSaving.lib. From that point, change-over will be performed automatically, allowing all functions to be executed properly and at the right time.

Note

**Error message in WAGO I/O CHECK is possible after a power failure!**

If you are using the software "WAGO-I/O-CHECK" after a loss of power has occurred, error messages may be generated. Should this occur, call up the Web-based management system and set the actual time under "Clock". Then, call up the "WAGO-I/O-CHECK" program again

Note

**Loss of telegrams possible when performing configuration during ongoing operation!**

Telegrams may be lost if configuration is performed using WAGO-I/O-CHECK while the system is in operation.

Note

**Use a WAGO RTC module for time synchronization!!**

You can use a WAGO 750-640 RTC Module for your node to utilize the actual encoded time (Real-time – RTC) in your higher-level control system.

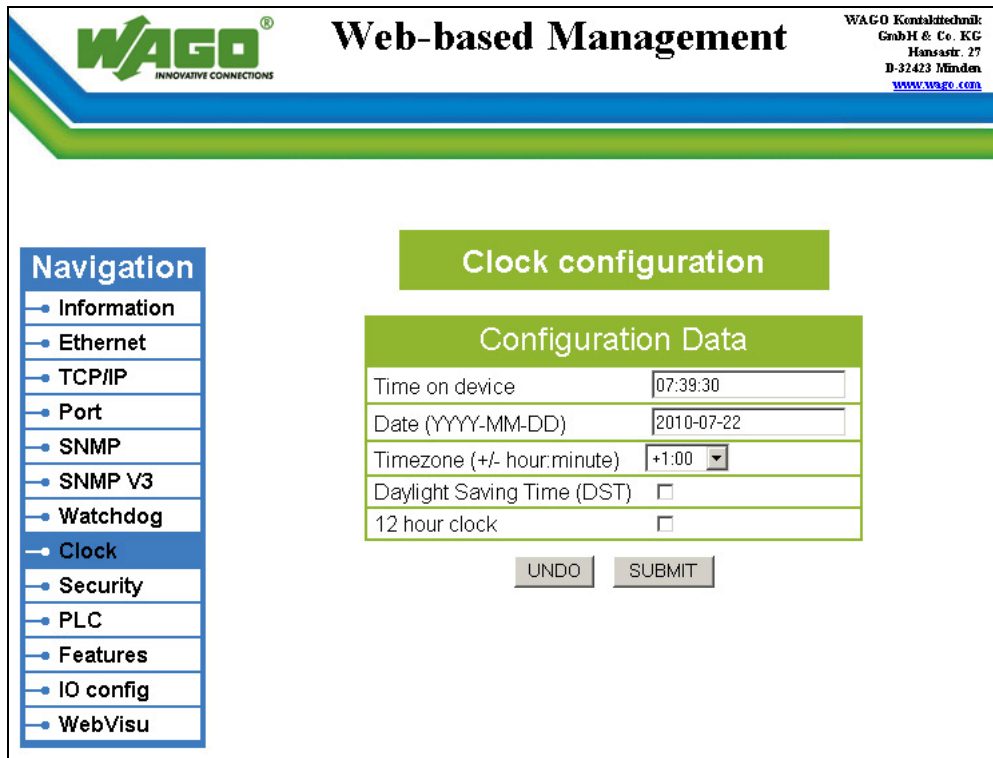


Figure 65: WBM page "Clock"

Table 45: WBM page "Clock"

Configuration Data			
Entry	Default	Value (example)	Description
Time on device	Coordinated Universal Time UTC	09:16:41	Set current time
Date (YYYY-MM-DD)	Date based on UTC	2009-05-06	Set current date
Time zone (+/- hour)	0	1 (MEZ)	Set time zone offset from the Coordinated Universal Time (UTC)
Daylight Saving Time (DST)/ Summer Time	Summer time <input checked="" type="checkbox"/>	Summer time <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Enable summer time
			<input type="checkbox"/> Enable winter time
12 hour clock	12 hour clock <input checked="" type="checkbox"/>	12 hour clock <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Enable 12-hour display
			<input type="checkbox"/> Enable 24-hour display

10.8 Security

Use the "Security" HTML page with passwords to set up read and/or write access for various user groups to protect against configuration changes.

Note



Passwords can only be changed by "admin" and after software reset!

The "admin" user and associated password are required to change passwords.

Press the [**Software Reset**] button to restart the software for the setting changes to take effect.

Note



Note password restrictions!

The following restrictions apply for passwords:

- Max. 16 characters
 - Letters and numbers only
 - No special characters or umlauts
-

Note



Renew access after software reset!

If you initiate a software reset on this page, then the fieldbus coupler/controller starts with the configurations previously loaded into the EEPROM and the connection to the browser is interrupted.

If you changed the IP address previously, you have to use the changed IP address to access the device from the browser.

You have have not changed the IP address and performed other settings, you can restor the connection by refreshing the browser.

WAGO
INNOVATIVE CONNECTIONS

Web-based Management

WAGO Kontakttechnik
GmbH & Co. KG
Hansastr. 27
D-32423 Minden
www.wago.com

Navigation

- Information
- Ethernet
- TCP/IP
- Port
- SNMP
- SNMP V3
- Watchdog
- Clock
- Security**
- PLC
- Features
- IO config
- WebVisu

Security

This page is intended to disable the basic authentication. Additionally you can set new passwords for the existing user. The new values are stored in an EEPROM and changes will take effect after the next software or hardware reset.

Webserver Security

Webserver authentication enabled

UNDO SUBMIT

Webserver and FTP User configuration

User: Password:

Confirm Password:

UNDO SUBMIT

Attention: You will lose the connection to the webserver after the software reset, if the IP configuration was changed. Please load the webpage with the proper address in this case again.

Software Reset

Figure 66: WBM page "Security"

Table 46: WBM page "Security"

Webserver Security		
Entry	Default	Description
Webserver authentication enabled	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Enable password protection to access the Web interface
		<input type="checkbox"/> Disable password protection to access the Web interface
Webserver and FTP User configuration ^{*)}		
Entry	Default	Description
User	guest	Select admin, guest or user
Password	guest	Enter password
Confirm password		Enter password again to confirm

^{*)} The following default groups exist:

User: admin	Password: wago
User: guest	Password: guest
User: user	Password: user

10.9 PLC Info

The WBM page „PLC Info “ contains information about the current CoDeSys project. First, these information has to be registered in the CoDeSys project under the menu „Project “→ „Projekt Information “.

The screenshot shows the WAGO Web-based Management System interface. At the top left is the WAGO logo with the tagline 'INNOVATIVE CONNECTIONS'. To the right is the company name 'WAGO Kontakttechnik GmbH & Co. KG' and address 'Hansastr. 27, D-32423 Minden', along with the website 'www.wago.com'. The page has a blue and green header. On the left is a 'Navigation' menu with items: Information, Ethernet, TCP/IP, Port, SNMP, SNMP V3, Watchdog, Clock, Security, **PLC Info** (highlighted), PLC, Features, IO config, Disk Info, and WebVisu. The main content area has a green header for 'CoDeSys Info' with the text: 'This page provides information about the current CoDeSys-Project.' Below this is another green header for 'Project-Info' followed by a table:

Title	
Version	
Date	14.03.2012 12:56:30
Description	
Author	
ID	348555

Figure 67: WBM page „PLC Info“

Table 47: WBM page „PLC Info“

Project Info			
Entry	Default value	Value (Example)	Description
Title	_____	SSL Client Example	Name of project
Version	_____	1.0.0	Project version
Date	_____	15.05.2012 08:50:27	Date and Time of the project
Description	_____	Testlient zum Aufbau von SSL-Verbindungen	Description
Author	_____	JW	Project author
ID	_____	70632	Project ID

10.10 PLC

Click the "PLC" link to access a Web site where you can define the PFC functionality settings for your controller.

WAGO Web-based Management

WAGO Kontakttechnik GmbH & Co. KG
Hansarstr. 27
D-32423 Minden
www.wago.com

Navigation

- Information
- Ethernet
- TCP/IP
- Port
- SNMP
- SNMP V3
- Watchdog
- Clock
- Security
- PLC**
- Features
- IO config
- WebVisu

PLC Configuration

This page is for the configuration of the PLC. The configuration is stored in an EEPROM. Changes of the process image setting will take effect after the next software or hardware reset. All other changes will take effect immediately. For more information see the manual.

PLC Features

Function	Description	Enabled
Process image	Set outputs to zero, if user program is stopped.	<input checked="" type="checkbox"/>
WebVisu	Set 'webvisu.htm' as default.	<input type="radio"/>
	Open 'webvisu.htm' in frame.	<input type="radio"/>
	Open 'webvisu.htm' in new window.	<input checked="" type="radio"/>
I/O configuration	Compatible handling for ea-config.xml	<input type="checkbox"/>
	Insert monitoring entries into ea-config.xml	<input type="checkbox"/>

UNDO SUBMIT

Figure 68: WBM page "PLC"

Note



Return to WBM view via the IP address of the fieldbus controller!

The "Webvisu.htm" page does not have any hyperlinks to the other Web sites. To deactivate this starting page function, or to go to other pages using hyperlinks, enter the IP address for your controller and the address for the original starting page in the URL line of your browser with the following syntax:

<http://IP address of your controller/webserv/Index.ssi>

Table 48: WBM page "PLC"

PLC Features		
Function	Default	Description
Process image	Set outputs to zero, if user program is stopped <input type="checkbox"/>	<input checked="" type="checkbox"/> Activate, if all outputs must be set at zero when stopping the user program
		<input type="checkbox"/> Disable, if all outputs must remain at the last current value when stopping the user program
WebVisu	Set 'webvisu.htm' as default <input type="radio"/>	<input checked="" type="radio"/> Activate, if the page "Webvisu.htm" must be opened as starting page when calling up WMB instead of the standard starting page "Status Information"
		<input type="radio"/> Activate, if the standard starting page "Status Information" must be opened when calling up WMB
	Open 'webvisu.htm' in frame <input type="radio"/>	<input checked="" type="radio"/> Activate, if the page "Webvisu.htm" must be opened in the same frame
		<input type="radio"/> Activate, if the page "Webvisu.htm" must be opened in another frame
	Open 'webvisu.htm' in new window <input checked="" type="radio"/>	<input checked="" type="radio"/> Activate, if the page "Webvisu.htm" must be opened in the same window
		<input type="radio"/> Activate, if the page "Webvisu.htm" must be opened in another window

Table: WBM page "PLC"

PLC Features														
Function	Default	Description												
I/O configuration	Compatible handling for ea-config.xml <input type="checkbox"/>	<p>Activate, if the write authorizations must be assigned to the outputs of all bus terminals based on an existing file "ea-config.xml".</p> <p>Here, note whether a control system configuration has already been created and, if so, whether this configuration is correct or incorrect (see the following table).</p> <p>The current process values are displayed on the website "IO config", in addition to the displayed data channels.</p> <p><input checked="" type="checkbox"/></p>												
		<p>Disable, if the write authorizations must be assigned to the outputs of all bus terminals of the PLC</p> <p>Here, note whether a control system configuration has already been created and, if so, whether this configuration is correct or incorrect (see the following table).</p> <p><input type="checkbox"/></p>												
		<table border="1"> <thead> <tr> <th></th> <th>I/O configuration (function activated)</th> <th>I/O configuration (function deactivated, standard setting):</th> </tr> </thead> <tbody> <tr> <td> <p>No control system configuration has been created in the project</p> </td> <td> <p>Writing privileges to the outputs of all modules are assigned on the basis of an existing ea-config.xml.</p> <p>The ea-config.xml file must be completely error-free; otherwise the writing privileges for all modules will be assigned to the standard fieldbus.</p> </td> <td> <p>The outputs for all modules are assigned to the PLC. Any ea-config.xml file that may already be present is ignored and overwritten.</p> </td> </tr> <tr> <td> <p>Correct control system configuration has been created in the project</p> </td> <td colspan="2"> <p>Writing privileges to the module outputs is taken from the control system configuration. A corresponding ea-config.xml file is generated in the file system.</p> </td> </tr> <tr> <td> <p>Incorrect control system configuration has been created in the project</p> </td> <td colspan="2"> <p>The standard fieldbus is granted writing privileges to the outputs of all the modules.</p> </td> </tr> </tbody> </table>		I/O configuration (function activated)	I/O configuration (function deactivated, standard setting):	<p>No control system configuration has been created in the project</p>	<p>Writing privileges to the outputs of all modules are assigned on the basis of an existing ea-config.xml.</p> <p>The ea-config.xml file must be completely error-free; otherwise the writing privileges for all modules will be assigned to the standard fieldbus.</p>	<p>The outputs for all modules are assigned to the PLC. Any ea-config.xml file that may already be present is ignored and overwritten.</p>	<p>Correct control system configuration has been created in the project</p>	<p>Writing privileges to the module outputs is taken from the control system configuration. A corresponding ea-config.xml file is generated in the file system.</p>		<p>Incorrect control system configuration has been created in the project</p>	<p>The standard fieldbus is granted writing privileges to the outputs of all the modules.</p>	
			I/O configuration (function activated)	I/O configuration (function deactivated, standard setting):										
	<p>No control system configuration has been created in the project</p>	<p>Writing privileges to the outputs of all modules are assigned on the basis of an existing ea-config.xml.</p> <p>The ea-config.xml file must be completely error-free; otherwise the writing privileges for all modules will be assigned to the standard fieldbus.</p>	<p>The outputs for all modules are assigned to the PLC. Any ea-config.xml file that may already be present is ignored and overwritten.</p>											
	<p>Correct control system configuration has been created in the project</p>	<p>Writing privileges to the module outputs is taken from the control system configuration. A corresponding ea-config.xml file is generated in the file system.</p>												
	<p>Incorrect control system configuration has been created in the project</p>	<p>The standard fieldbus is granted writing privileges to the outputs of all the modules.</p>												
<p>Correct control system configuration has been created in the project</p>														
<p>Incorrect control system configuration has been created in the project</p>														
Insert monitoring entries into ea-config.xml <input type="checkbox"/>	<p>Activate to also display the current process values on the html page "IO config" for the displayed data channels.</p> <p><input checked="" type="checkbox"/></p> <p>Disable, if no process values must be displayed on the html page "IO config".</p> <p><input type="checkbox"/></p>													

10.11 Features

Use the "Features" HTML page to enable or disable additional functions.

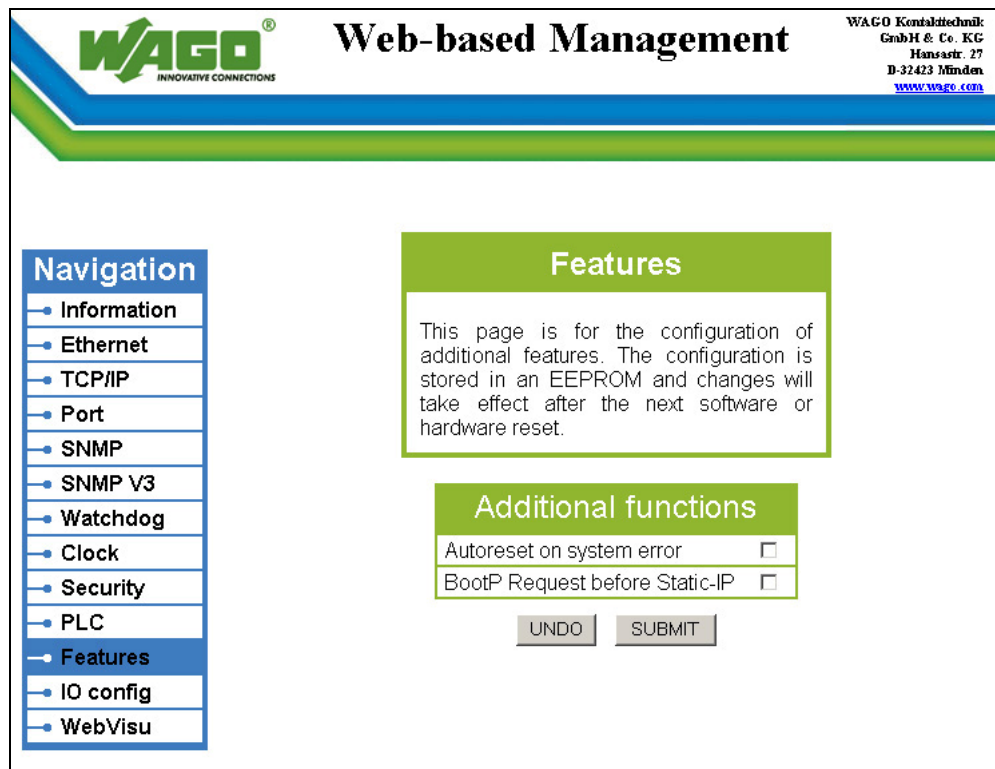


Figure 69: WBM page "Features"

Table 49: WBM page "Features"

Additional functions		
Entry	Default	Description
Autoreset on system error	<input type="checkbox"/>	<input checked="" type="checkbox"/> enables an automatic software reset to be conducted when a system error occurs
		<input type="checkbox"/> disables an automatic software reset to be conducted when a system error occurs
BootP Request before Static-IP	<input type="checkbox"/>	<input checked="" type="checkbox"/> Automatically set the static IP address enabled. For this configuration, the fieldbus coupler/controller uses a statically configured IP address if the request via BootP fails.
		<input type="checkbox"/> Automatically set the static IP address disabled. For this configuration, the IP address request via BootP is repeated in the event of error.

10.12 I/O Config

Click the link "I/O config" to view the configuration and/or write access privileges for the outputs of your fieldbus node.

The node structure created using the "WAGO-I/O-PRO I/O Configurator" hardware configuration tool is displayed in the window. If no modules are shown in this window, no hardware configuration and, thus, no allocation of write access privileges have been assigned. In this case, the handling defined at the Web site "PLC" by the function "I/O configuration - Compatible handling for ea-config.xml" will be applied to assign the write privileges for all outputs either to the standard fieldbus, or to the PLC.

The screenshot shows the WAGO Web-based Management interface. At the top, there is a header with the WAGO logo and contact information. The main content area is titled "I/O configuration" and contains a "Configuration details" section with the following data:

Number of modules on terminalbus	4
Number of modules in I/O configuration	0

Below this is a section for "I/O configuration file" with a table structure:

Pos	Module	Type	Mapping
-----	--------	------	---------

Figure 70: WBM page "I/O Config"



Information

Additional Information

For more detailed information about the WAGO-I/O-PRO I/O Configurator, refer to the Section "Startup of Fieldbus Node".

When the function "I/O configuration Insert monitoring entries into ea-config.xml" is also activated at the Web site "PLC", the current process values will also be shown for the data channels that are displayed.

Table 50: WBM page "I/O configuration"

Configuration details		
Entry	Value (Example)	Description
Number of modules on terminalbus	5	Number of I/O modules (hardware)
Number of modules in I/O configuration	5	Number of I/O modules in the hardware configuration of the I/O
I/O configuration file		
Entry	Value (Example)	Description
Pos	1	Position of the I/O module in the hardware
Module	750-4xx M001Ch1 M001Ch2	Product number of the integrated I/O module M = module, 001 = position 1, Ch1 = channel 1 M = module, 002 = position 2, Ch2 = channel 2
Type	2DI	I/O module type, e.g. 2 DI (2 Channel Digital Input Module)
Mapping	Fieldbus 3	Mapping via PLC, fieldbus 1 etc. (Entries depend on the coupler/controller, see WAGO-I/O-PRO under control parameters/module parameters)

Note



Enter I/O modules in the I/O Configurator!

Enter the I/O modules used in the I/O configurator of WAGO-I/O-PRO. Here, open the **Control Configuration** in the **Resources** register and add your I/O modules to the I/O module figure.

The added I/O modules must match the hardware in sequence and quantity. The entries "Number of modules on terminalbus" and "Number of modules in I/O configuration" on the html page "PLC" serve as control.

10.13 WebVisu

The visualization of your programmed application is displayed on the html page "WebVisu", provided you have created it with the visualization editor in WAGO-I/O-PRO and loaded it into the controller.

Perform the following settings in WAGO-I/O-PRO, so that an html page with your visualization is automatically created at the transmission of your project:

1. Double click to open the **Target System Settings** in the **Resource** register.
2. Open the **Visualization** register.
3. Select the **Web Visualization** option with a hatch mark.
4. Confirm with **OK**.

A link is then created to this html page "WebVisu" by the Web-based Management system. You can set the html page "WebVisu" as the starting page.

1. Call up the page "PLC" in the web-based Management-System.
2. a.) To set the HTML page "WebVisu" as the start page, use the function **WebVisu – Set 'webvisu.htm' as default**. When accessing the web-based management system, the "WebVisu" page is opened instead of the default WBM start page "Information". However, the links to switch to the other WBM pages is then no longer available.

Note



Returning to the "WebVisu.htm" page is only possible via the IP address of the fieldbus controller!

The "Webvisu.htm" page does not have any hyperlinks to other Web sites. To deactivate the starting page function again, or to go to other pages using hyperlinks, enter the IP address for your controller and the address for the original starting page in the URL line of your browser with the following syntax:

<http://IP address of your controller/webserv/Index.ssi>.

- b.) To call up the the HTML page "WebVisu" in an eternal window (default setting), use the function **WebVisu – Open 'webvisu.htm' in new window**. Clicking on the "WebVisu" link opens a new window that displays the HTML page with visualization of your configured application. The links to switch to the other WBM pages are still available with this setting.
- c.) To call up the HTML page "WebVisu" on the WBM site directly, use the function **WebVisu – Open 'webvisu.htm' in frame**. Clicking on the "WebVisu" link opens the HTML page with visualization of your configured application in a frame in the WBM window directly. The

links to switch to the other WBM pages are still available with this setting.

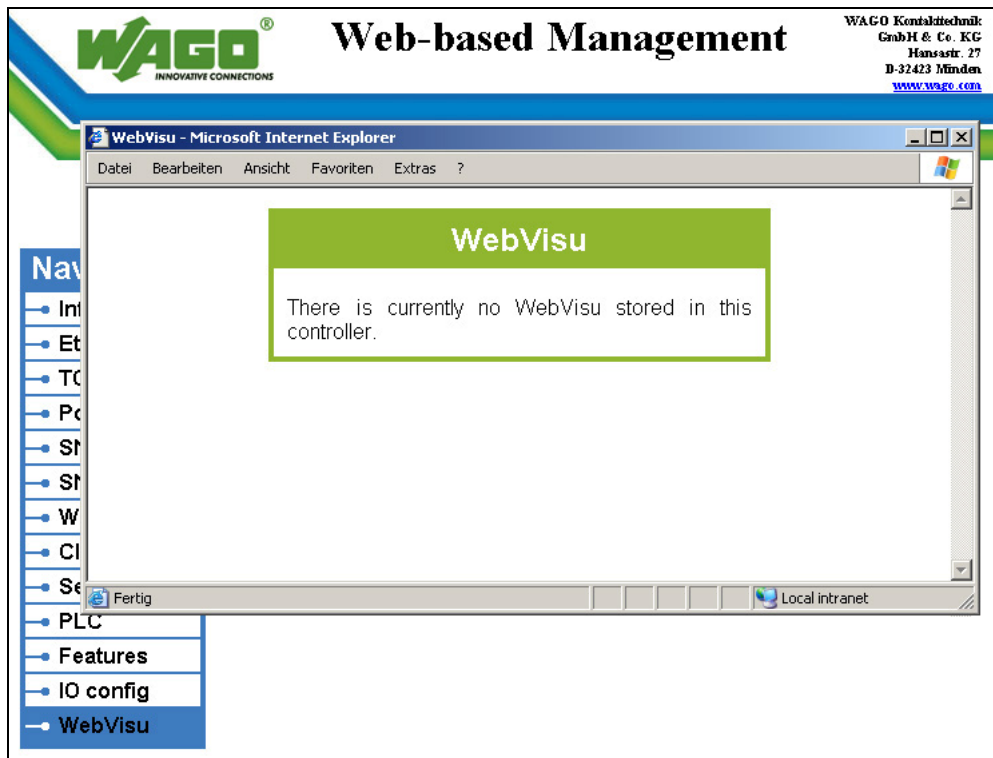


Figure 71: WBM page "WebVisu"

11 Diagnostics

11.1 LED Signaling

For on-site diagnostics, the fieldbus controller has several LEDs that indicate the operational status of the controller or the entire node (see following figure).

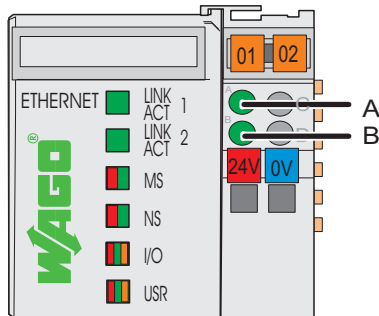


Figure 72: Display Elements

The diagnostics displays and their significance are explained in detail in the following chapter.

The LEDs are assigned in groups to the various diagnostics areas:

Table 51: LED assignment for diagnostics

Diagnostics area	LEDs
Fieldbus status	<ul style="list-style-type: none"> • LINK ACT Port 1 • LINK ACT Port 2 • MS • NS
Node status	<ul style="list-style-type: none"> • I/O • USR
Status Supply Voltage	<ul style="list-style-type: none"> • A (system supply) • B (field supply)

11.1.1 Evaluating Fieldbus Status

The health of the ETHERNET Fieldbus is signaled through the top LED group ('LINK ACT 1, 2', 'MS', and 'NS').

The two-colored LEDs 'MS' (module status) and 'NS' (network status) are solely used by the Ethernet/IP protocol. These two LEDs conform to the Ethernet/IP specifications.

Table 52: Fieldbus diagnostics – solution in event of error

LED Status	Meaning	Solution
LINK ACT 1, 2		
green	The fieldbus node is connected to the physical network.	-
green flashing	The fieldbus node sends and receives Ethernet telegrams	-
off	The fieldbus node is not connected to the physical network.	1. Check the fieldbus cable.
MS		
green	Normal operation	-
green flashing	The system is not yet configures	-
red	The system indicates a not remediable error	1. Restart the device by turning the power supply off and on again. 2. If the error still exists, please contact the I/O support.
red/green flashing	Self test	-
off	No system supply voltage	1. Check the supply voltage.
NS		
green	At least one connection (MODBUS/TCP or Ethernet/IP) is developed (also connection to the Message rout applies)	-
grün flashing	No connection (MODBUS/TCP or Ethernet/IP).	-
red	The system indicates a double IP-address in the network	1. Use an IP address that is not used yet.
red flashing	At least one connection (MODBUS/TCP or Ethernet/IP) announced a Timeout, where the controller functions as target.	1. Restart the device by turning the power supply off and on again. 2. Develop a new connection.
red/green flashing	Self test	-
off	No IP address is assigned to the system.	1. Assign to the system an IP address for example by BootP or DHCP.

11.1.2 Evaluating Node Status - I/O LED (Blink Code Table)

The communication status between fieldbus coupler/controller is indicated by the I/O LED.

Table 53: Node status diagnostics – solution in event of error

LED Status	Meaning	Solution
I/O		
green	The fieldbus node is operating correctly.	Normal operation.
orange flashing	The internal data bus is initialized, 1-2 seconds of rapid flashing indicate start-up.	-
red	Controller hardware defect	Replace the fieldbus coupler/controller.
red flashing	General internal bus error	Note the following blinking sequence.
red cyclical flashing	Up to three successive blinking sequences indicate internal data bus errors. There are short intervals between the sequences.	Evaluate the blinking sequences based on the following blink code table. The blinking indicates an error message comprised of an error code and error argument.
off	No data cycle on the internal bus.	The fieldbus coupler/controller supply is off.

Device boot-up occurs after turning on the power supply. The I/O LED is orange.

After a trouble-free start-up, the I/O LED is green.

In the event of an error, the I/O LED continues to blink red. Blink codes indicate detailed error messages. An error is indicated cyclically by up to 3 blinking sequences.

After elimination of the error, restart the node by turning the power supply of the device off and on again.

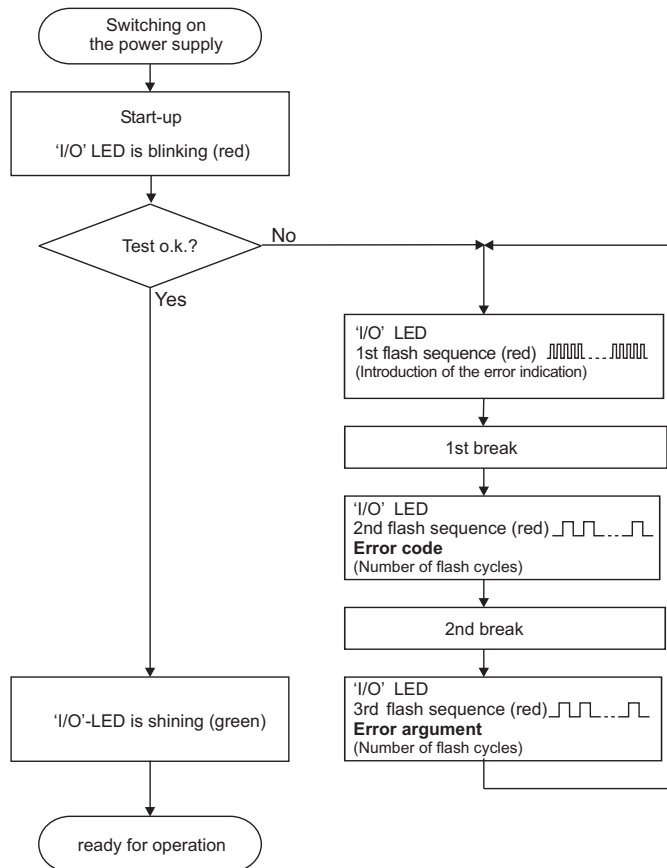


Figure 73: Node status - I/O LED signaling

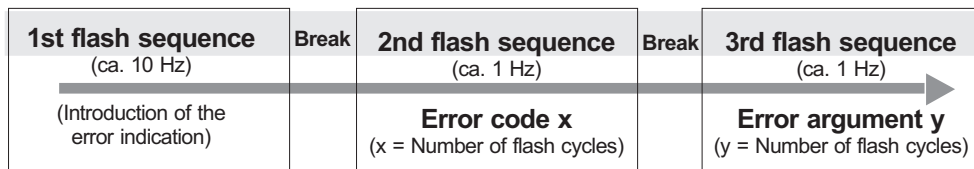


Figure 74: Error message coding

Example of a module error:

- The I/O LED starts the error display with the first blinking sequence (approx. 10 Hz).
- After the first break, the second blinking sequence starts (approx. 1 Hz):
The I/O LED blinks four times.
Error code 4 indicates "data error internal data bus".
- After the second break, the third blinking sequence starts (approx. 1 Hz):
The I/O LED blinks twelve times.
Error argument 12 means that the internal data bus is interrupted behind the twelfth I/O module.

The thirteenth I/O module is either defective or has been pulled out of the assembly.

Table 54: Blink code- table for the I/O LED signaling, error code 1

Error code 1: "Hardware and configuration error"		
Error Argument	Error Description	Solution
1	Overflow of the internal buffer memory for the attached I/O modules.	<ol style="list-style-type: none"> 1. Turn off the power for the node. 2. Reduce the number of I/O modules and turn the power supply on again. 3. If the error persists, replace the fieldbus controller.
2	I/O module(s) with unknown data type	<ol style="list-style-type: none"> 1. Determine the faulty I/O module by first turning off the power supply. 2. Plug the end module into the middle of the node. 3. Turn the power supply on again. 4. - LED continues to flash? - Turn off the power supply and plug the end module into the middle of the first half of the node (toward the fieldbus controller). - LED not flashing? - Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller). 5. Turn the power supply on again. 6. Repeat the procedure described in step 4 while halving the step size until the faulty I/O module is detected. 7. Replace the faulty I/O module. 8. Inquire about a firmware update for the fieldbus controller.
3	Invalid check sum in the parameter area of the fieldbus controller.	<ol style="list-style-type: none"> 1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
4	Fault when writing in the serial EEPROM.	<ol style="list-style-type: none"> 1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
5	Fault when reading the serial EEPROM	<ol style="list-style-type: none"> 1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
6	The I/O module configuration after AUTORESET differs from the configuration determined the last time the fieldbus controller was powered up.	<ol style="list-style-type: none"> 1. Restart the fieldbus controller by turning the power supply off and on.
7	Invalid hardware-firmware combination.	<ol style="list-style-type: none"> 1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.

Table 54: Blink code- table for the I/O LED signaling, error code 1

Error code 1: "Hardware and configuration error"		
Error Argument	Error Description	Solution
8	Timeout during serial EEPROM access.	<ol style="list-style-type: none"> 1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
9	Bus controller initialization error	<ol style="list-style-type: none"> 1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
10	Buffer power failure real-time clock (RTC)	<ol style="list-style-type: none"> 1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
11	Fault during read access to the real-time clock (RTC)	<ol style="list-style-type: none"> 1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
12	Fault during write access to the real-time clock (RTC)	<ol style="list-style-type: none"> 1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
13	Clock interrupt fault	<ol style="list-style-type: none"> 1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
14	Maximum number of gateway or mailbox modules exceeded	<ol style="list-style-type: none"> 1. Turn off the power for the node. 2. Reduce the number of corresponding modules to a valid number.

Table 55: Blink code table for the I/O LED signaling, error code 2

Error code 2: -not used-		
Error Argument	Error Description	Solution
-	Not used	-

Table 56: Blink code table for the I/O LED signaling, error code 3

Error code 3: "Protocol error, internal bus"		
Error Argument	Error Description	Solution
-	Internal data bus communication is faulty, defective module cannot be identified.	<p>- Are passive power supply modules (750-613) located in the node? -</p> <ol style="list-style-type: none"> 1. Check that these modules are supplied correctly with power. 2. Determine this by the state of the associated status LEDs. <p>- Are all modules connected correctly or are there any 750-613 Modules in the node? -</p> <ol style="list-style-type: none"> 1. Determine the faulty I/O module by turning off the power supply. 2. Plug the end module into the middle of the node. 3. Turn the power supply on again. 4. - LED continues to flash? - Turn off the power supply and plug the end module into the middle of the first half of the node (toward the fieldbus controller). - LED not flashing? - Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller). 5. Turn the power supply on again. 6. Repeat the procedure described in step 4 while halving the step size until the faulty I/O module is detected. 7. Replace the faulty I/O module. 8. Inquire about a firmware update for the fieldbus controller.

Table 57: Blink code table for the I/O LED signaling, error code 4

Error code 4: "Physical error, internal bus"		
Error Argument	Error Description	Solution
-	Internal bus data transmission error or interruption of the internal data bus at the fieldbus controller	<ol style="list-style-type: none"> 1. Turn off the power supply to the node. 2. Plug in an end module behind the fieldbus controller. 3. Turn the power supply on. 4. Observe the error argument signaled. <p>- Is no error argument indicated by the I/O LED? -</p> <ol style="list-style-type: none"> 5. Replace the fieldbus controller. <p>- Is an error argument indicated by the I/O LED? -</p> <ol style="list-style-type: none"> 5. Identify the faulty I/O module by turning off the power supply. 6. Plug the end module into the middle of the node. 7. Turn the power supply on again. 8. - LED continues to flash? - Turn off the power and plug the end module into the middle of the first half of the node (toward the fieldbus controller). - LED not flashing? - Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller). 9. Turn the power supply on again. 10. Repeat the procedure described in step 6 while halving the step size until the faulty I/O module is detected. 11. Replace the faulty I/O module. 12. If there is only one I/O module on the fieldbus controller and the LED is flashing, either the I/O module or fieldbus controller is defective. Replace the defective component.
n*	Interruption of the internal data bus behind the nth bus module with process data	<ol style="list-style-type: none"> 1. Turn off the power supply to the node. 2. Replace the (n+1) I/O module containing process data. 3. Turn the power supply on.

* The number of light pulses (n) indicates the position of the I/O module.
I/O modules without data are not counted (e.g., supply modules without diagnostics)

Table 58: Blink code table for the I/O LED signaling, error code 5

Error code 5: "Initialization error, internal bus"		
Error Argument	Error Description	Solution
n*	Error in register communication during internal bus initialization	<ol style="list-style-type: none"> 1. Turn off the power supply to the node. 2. Replace the (n+1) I/O module containing process data. 3. Turn the power supply on.

* The number of light pulses (n) indicates the position of the I/O module.
I/O modules without data are not counted (e.g., supply modules without diagnostics)

Table 59: Blink code- table for the I/O LED signaling, error code 6

Error code 6: "Fieldbus specific errors"		
Error Argument	Error description	Solution
1	Invalid MACID	1. Turn off the power supply of the node. 2. Exchange fieldbus controller. 3. Turn the power supply on again.
2	Ethernet Hardware initialization error	1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, exchange the fieldbus controller.
3	TCP/IP initialization error	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. If the error still exists, exchange the bus coupler.
4	Network configuration error (no IP Address)	1. Check the settings of BootP server.
5	Application protocol initialization error	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. If the error still exists, exchange the bus coupler.
6	Process image is too large	1. Turn off the power supply of the node. 2. Reduce number of I/O modules
7	Double IP address in network	1. Change configuration. Use another IP address, which is not yet present in network. 2. Restart the fieldbus coupler by turning the power supply off and on again.
8	Error when building the process image	1. Turn off the power supply of the node. 2. Reduce number of I/O modules 3. Restart the fieldbus coupler by turning the power supply off and on again. 4. If the error still exists, exchange the bus coupler.
9	Error with mapping between bus modules and fieldbus	1. Check EA-Config.xml file on the fieldbus controller

Table 60: Blink code table for the 'I/O' LED signaling, error code 7...9

Error code 7...9: -not used-		
Error Argument	Error Description	Solution
-	Not used	

Table 61: Blink code table for the 'I/O' LED signaling, error code 10

Error code 10: "PLC program fault"		
Error Argument	Error Description	Solution
1	Error when implementing the PFC run time system	<ol style="list-style-type: none"> 1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, please contact the I/O Support.
2	Error when generating the PFC inline code	<ol style="list-style-type: none"> 1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, please contact the I/O Support.
3	An IEC task exceeded the maximum running time or the sampling interval of the IEC task could not be kept (Watchdog)	<ol style="list-style-type: none"> 1. Check the task configuration concerning the adjusted sampling intervals and watchdog times.
4	PFC Web-Visualization initialization error	<ol style="list-style-type: none"> 1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, please accomplish a reset (origin) in WAGO-I/O-PRO. 3. Compile the project again. 4. Transfer the project to the controller.
5	Error when synchronizing the PLC configuration with the internal data bus	<ol style="list-style-type: none"> 1. Check the information of the connected modules in the PLC configuration of WAGO-I/O-PRO. 2. Compare this information with the modules that are actually connected. 3. Compile the project again. 4. Transfer the project to the controller.

Table 62: Blink code table for the 'I/O' LED signaling, error code 11

Error code 11: "Gateway-/Mailbox I/O module fault"		
Error Argument	Error Description	Solution
1	Maximum number of Gateway modules exceeded	<ol style="list-style-type: none"> 1. Turn off the power supply of the node. 2. Reduce number of Gateway modules. 3. Turn the power supply on again.
2	Maximum size of Mailbox exceeded	<ol style="list-style-type: none"> 1. Turn off the power supply of the node. 2. Reduce the Mailbox size. 3. Turn the power supply on again.
3	Maximum size of process image exceeded due to the put Gateway modules	<ol style="list-style-type: none"> 1. Turn off the power supply of the node. 2. Reduce the data width of the Gateway modules. 3. Turn the power supply on again.

* The number of blink pulses (n) indicates the position of the I/O module.
I/O modules without data are not counted (e.g. supply module without diagnosis)

11.1.2.1 USR LED

The bottom indicator LED ("USR") is provided for visual output of information.

Control of the LED from the application program is conducted using the functions from the WAGO-I/O-PRO library "Visual.lib."

11.1.3 Evaluating Power Supply Status

The power supply unit of the device has two green LEDs that indicate the status of the power supply.

LED 'A' indicates the 24 V supply of the coupler.

LED 'B' or 'C' reports the power available on the power jumper contacts for field side power.

Table 63: Power supply status diagnostics – solution in event of error

LED Status	Meaning	Solution
A		
Green	Operating voltage for the system is available.	-
Off	No power is available for the system	Check the power supply for the system (24V and 0V).
B or C		
Green	The operating voltage for power jumper contacts is available.	-
Off	No operating voltage is available for the power jumper contacts.	Check the power supply for the power jumper contacts (24V and 0V).

11.2 Fault Behavior

11.2.1 Loss of Fieldbus

A fieldbus and, hence, a link failure is recognized when the set reaction time for the watchdog expires without initiation by the higher-order control system. This may occur, for example, when the Master is switched off, or when there is a disruption in the bus cable. An error at the Master can also result in a fieldbus failure. No connection via ETHERNET.

The MODBUS watchdog monitors the ongoing MODBUS communication via MODBUS protocol. A fieldbus failure is signaled by the red "I/O" LED lighting up, provided the MODBUS watchdog has been configured and activated.

Fieldbus monitoring independently of a certain protocol is possible using the function block 'FBUS_ERROR_INFORMATION' in the library "Mod_com.lib". This checks the physical connection between modules and the controller and assumes evaluation of the watchdog register in the control system program. The I/O bus remains operational and the process images are retained. The control system program can also be processed independently.

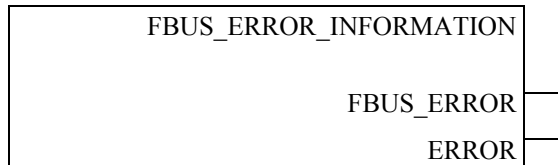


Figure 75: Function block for determining loss of fieldbus, independently of protocol

'FBUS_ERROR' (BOOL)	= FALSE	= no fault
	= TRUE	= loss of field bus
'ERROR' (WORD)	= 0	= no fault
	= 1	= loss of field bus

The node can be put into a safe status in the event of a fieldbus failure with the aid of these function block outputs and an appropriately programmed control system program.

Information



Loss of fieldbus detection through MODBUS protocol:

For detailed information about the watchdog register, refer to Section "MODBUS Functions", in particular Section "Watchdog (Fieldbus failure)".

Protocol-independent detection of loss of fieldbus:

The library 'Mod_com.lib' with function block 'FBUS_ERROR_INFORMATION' is normally included in the setup for the WAGO-I/O-PRO. You can integrate the library via register "Resources" at the bottom on the left of the workspace. Click **Insert** and then **Other libraries**. The Mod_com.lib is located in folder C:\Programme\WAGO Software\CoDeSys V2.3\Targets\WAGO\Libraries\32_Bit

11.2.2 Internal Data Bus Failure

I/O LED indicates an internal bus failure.

I/O LED flashed red:

When an internal data bus failure occurs, the fieldbus controller generates an error message (error code and error argument).

An internal data bus failure occurs, for example, if an I/O module is removed.

If the error occurs during operation, the output modules operate as they do during an internal data bus stop.

If the internal data bus error is resolved, the controller starts up after turning the power off and on similar to that of a normal start-up. The process data is transmitted again and the outputs of the node are set accordingly.

If the 'KBUS_ERROR_INFORMATION' function block is evaluated in the control program, then the 'ERROR', 'BITLEN', 'TERMINALS' and 'FAILADDRESS' output values are relevant.

'ERROR'	= FALSE	= No fault
('BITLEN'		= Bit length of the internal bus shift register
'TERMINALS'		= Number of I/O modules)
'ERROR'	= TRUE	= Internal Bus Error
('BITLEN'		= 0
'TERMINALS'		= 0)
'FAILADDRESS'		= Position of the I/O module after which the internal bus interruption arose, similar to the flashed error argument of the I/O LED

12 Fieldbus Communication

Fieldbus communication between master application and a WAGO fieldbus coupler/controller based on the ETHERNET standard normally occurs via an implemented fieldbus-specific application protocol.

Depending on the application, this can be e.g., MODBUS/TCP (UDP), EtherNet/IP, BACnet/IP, KNXnet/IP, PROFINET, SERCOS III or other.

In addition to the ETHERNET standard and the fieldbus-specific application protocol, there are also other communications protocols important for reliable communication and data transmission and other related protocols for configuring and diagnosing the system implemented in the WAGO fieldbus coupler/controller based on ETHERNET.

These protocols are explained in more detail in the other sections.

12.1 Implemented Protocols

12.1.1 Communication Protocols

12.1.1.1 IP (Internet Protocol)

The Internet protocol divides datagrams into segments and is responsible for their transmission from one network subscriber to another. The stations involved may be connected to the same network or to different physical networks which are linked together by routers.

Routers are able to select various paths (network transmission paths) through connected networks, and bypass congestion and individual network failures. However, as individual paths may be selected which are shorter than other paths, datagrams may overtake each other, causing the sequence of the data packets to be incorrect.

Therefore, it is necessary to use a higher-level protocol, for example, TCP to guarantee correct transmission.

IP Packet

In addition to the data units to be transported, the IP data packets contain a range of address information and additional information in the packet header.

Table 64: IP Packet

IP Header	IP Data
-----------	---------

The most important information in the IP header is the IP address of the transmitter and the receiver and the transport protocol used.

IP Addresses

To allow communication over the network each fieldbus node requires a 32 bit Internet address (IP address).



Note

IP Address must be unique!

For error free operation, the IP address must be unique within the network.

As shown below there are various address classes with net identification (net ID) and subscriber identification (subscriber ID) of varying lengths. The net ID defines the network in which the subscriber is located. The subscriber ID identifies a particular subscriber within this network.

Networks are divided into various network classes for addressing purposes:

- **Class A:** (Net ID: Byte 1, Host ID: Byte 2... Byte 4)

Table 65: Network Class A

e. g.	101	16	232	22
	01100101	00010000	11101000	00010110
0	Net ID	Host ID		

The highest bit in Class A networks is always '0'. This means the highest byte can be in a range of '0 0000000' to '0 1111111'.

Therefore, the address range of a Class A network in the first byte is always between 0 and 127.

- **Class B:** (Net ID: Byte 1 ... Byte 2, Host ID: Byte 3... Byte 4)

Table 66: Network Class B

e. g.	181	16	232	22
	10110101	00010000	11101000	00010110
10	Net ID		Host ID	

The highest bits in Class B networks are always '10'. This means the highest byte can be in a range of '10 000000' to '10 111111'.

Therefore, the address range of Class B networks in the first byte is always between 128 and 191.

- **Class C:** (Net ID: Byte 1 ... Byte 3, Host ID: Byte 4)

Table 67: Network Class C

e. g.	201	16	232	22
	11000101	00010000	11101000	00010110
110	Net ID			Host ID

The highest bits in Class C networks are always '110'. This means the highest byte can be in a range of '110 00000' to '110 11111'.

Therefore, the address range of Class C networks in the first byte is always between 192 and 223.

- **Additional network classes (D, E):** are only used for special tasks.

Key Data

Table 68: Key Data Class A, B and C

Network Class	Address range of the subnetwork	Possible number of	
		Networks	Hosts per Network
Class A	1.XXX.XXX.XXX ... 126.XXX.XXX.XXX	127 (2^7)	Approx. 16 Million (2^{24})
Class B	128.000.XXX.XXX ... 191.255.XXX.XXX	Approx. 16 Thousand (2^{14})	Ca. 65 Thousand (2^{16})
Class C	192.000.000.XXX ... 223.255.255.XXX	Approx. 2 Million (2^{21})	254 (2^8)

Each WAGO ETHERNET fieldbus coupler or controller can be easily assigned an IP address via the implemented BootP protocol. For small internal networks we recommend selecting a network address from Class C.

Note



Do not set IP addresses to 0.0.0.0 or 255.255.255.255!

Never set all bits to equal 0 or 1 in one byte (byte = 0 or 255). These are reserved for special functions and may not be allocated. Therefore, the address 10.0.10.10 may not be used due to the 0 in the second byte.

If a network is to be directly connected to the Internet, only registered, internationally unique IP addresses allocated by a central registration service may be used. These are available from InterNIC (International Network Information Center).

Note



Internet access only by the authorized network administrator!

Direct connection to the Internet should only be performed by an authorized network administrator and is therefore not described in this manual.

Subnets

To allow routing within large networks a convention was introduced in the specification RFC 950. Part of the Internet address, the subscriber ID is divided up again into a subnetwork number and the station number of the node. With the aid of the network number it is possible to branch into internal subnetworks within the partial network, but the entire network is physically connected together. The size and position of the subnetwork ID are not defined; however, the size is

dependent upon the number of subnets to be addressed and the number of subscribers per subnet.

Table 69: Class B Address with Field for Subnet IDs

1		8		16		24		32	
1	0	...	Network ID		Subnet ID		Host ID		

Subnet Mask

A subnet mask was introduced to encode the subnets in the Internet. This involves a bit mask, which is used to mask out or select specific bits of the IP address. The mask defines the subscriber ID bits used for subnet coding, which denote the ID of the subscriber. The entire IP address range theoretically lies between 0.0.0.0 and 255.255.255.255. Each 0 and 255 from the IP address range are reserved for the subnet mask.

The standard masks depending upon the respective network class are as follows:

- **Class A Subnet mask:**

Table 70: Subnet mask for Class A network

255	.0	.0	.0
-----	----	----	----

- **Class B Subnet mask:**

Table 71: Subnet mask for Class B network

255	.255	.0	.0
-----	------	----	----

- **Class C Subnet mask:**

Table 72: Subnet mask for Class C network

255	.255	.255	.0
-----	------	------	----

Depending on the subnet division the subnet masks may, however, contain other values beyond 0 and 255, such as 255.255.255.128 or 255.255.255.248.

Your network administrator allocates the subnet mask number to you.

Together with the IP address, this number determines which network your PC and your node belongs to.

The recipient node, which is located on a subnet initially, calculates the correct network number from its own IP address and the subnet mask.

Only then does it check the node number and delivers the entire packet frame, if it corresponds.

Table 73: Example for an IP address from a Class B network

IP address	172.16.233.200	10101100 00010000 11101001 11001000
Subnet mask	255.255.255.128	11111111 11111111 11111111 10000000
Net ID	172.16.0.0	10101100 00010000 00000000 00000000
Subnet ID	0.0.233.128	00000000 00000000 11101001 10000000
Host ID	0.0.0.72	00000000 00000000 00000000 01001000



Note

Specification of the network mask necessarily!

Specify the network mask defined by the administrator in the same way as the IP address when installing the network protocol.

Gateway

The subnets of the Internet are normally connected via gateways. The function of these gateways is to forward packets to other networks or subnets.

This means that in addition to the IP address and network mask for each network card, it is necessary to specify the correct IP address of the standard gateway for a PC or fieldbus node connected to the Internet. You should also be able to obtain this IP address from your network administrator.

The IP function is limited to the local subnet if this address is not specified.

RAW IP

Raw IP manages without protocols such as PPP (point-to-point protocol). With RAW IP, the TCP/IP packets are directly exchanged without handshaking, thus enabling the connection to be established more quickly.

However, the connection must beforehand have been configured with a fixed IP address. The advantages of RAW IP are high data transfer rate and good stability.

IP Multicast

Multicast refers to a method of transmission from a point to a group, which is a point-to-multipoint transfer or multipoint connection. The advantage of multicast is that messages are simultaneously transferred to several users or closed user groups via one address.

IP multicasting at the Internet level is realized with the help of the Internet Group Message Protocol IGMP; neighboring routers use this protocol to inform each other on membership to the group.

For distribution of multicast packets in the sub-network, IP assumes that the datalink layer supports multicasting. In the case of Ethernet, you can provide a packet with a multicast address in order to send the packet to several recipients with a single send operation. Here, the common medium enables packets to be sent simultaneously to several recipients. The stations do not have to inform each other on who belongs to a specific multicast address – every station physically receives every packet. The resolution of IP address to Ethernet address is solved by the use of algorithms, IP multicast addresses are embedded in Ethernet multicast addresses.

12.1.1.2 TCP (Transmission Control Protocol)

As the layer above the Internet protocol, TCP (Transmission Control Protocol) guarantees the secure transport of data through the network.

TCP enables two subscribers to establish a connection for the duration of the data transmission. Communication takes place in full-duplex mode (i.e., transmission between two subscribers in both directions simultaneously).

TCP provides the transmitted message with a 16-bit checksum and each data packet with a sequence number.

The receiver checks that the packet has been correctly received on the basis of the checksum and then sets off the sequence number. The result is known as the acknowledgement number and is returned with the next self-sent packet as an acknowledgement.

This ensures that the lost TCP packets are detected and resent, if necessary, in the correct sequence.

TCP Data Packet

The packet header of a TCP data packet is comprised of at least 20 bytes and contains, among others, the application port number of the transmitter and the receiver, the sequence number and the acknowledgement number.

The resulting TCP packet is used in the data unit area of an IP packet to create a TCP/IP packet.

TCP Port Numbers

TCP can, in addition to the IP address (network and subscriber address), respond to a specific application (service) on the addressed subscriber. For this the applications located on a subscriber, such as a web server, FTP server and others are addressed via different port numbers. Well-known applications are assigned fixed ports to which each application can refer when a connection is built up (Examples: Telnet Port number: 23, http Port number: 80).

A complete list of "standardized services" is contained in the RFC 1700 (1994) specifications.

12.1.1.3 UDP (User Datagram Protocol)

The UDP protocol, like the TCP protocol, is responsible for the transport of data. Unlike the TCP protocol, UDP is not connection-orientated; meaning that there are no control mechanisms for the data exchange between transmitter and receiver. The advantage of this protocol is the efficiency of the transmitted data and the resulting higher processing speed.

12.1.2 Configuration and Diagnostics Protocols

12.1.2.1 BootP (Bootstrap Protocol)

The "Bootstrap Protocol" (BootP) can be used to assign an IP address and other parameters to the fieldbus coupler/controller in a TCP/IP network. Subnet masks and gateways can also be transferred using this protocol. Protocol communication is comprised of a client request from the fieldbus coupler or controller and a server response from the PC.

A broadcast request is transmitted to Port 67 (BootP server) via the protocol that contains the hardware address (MAC ID) for the fieldbus coupler or controller.

The BootP server then receives this message. The server contains a database in which the MAC ID and IP addresses are assigned to one another. When a MAC address is found a broadcast reply is transmitted via the network.

The fieldbus coupler/controller "listens" at the specified Port 68 for a response from the BootP server. Incoming packets contain information such as the IP address and the MAC address for the fieldbus coupler/controller. A fieldbus coupler/controller recognizes by the MAC address that the message is intended for that particular fieldbus coupler/controller and accepts the transmitted IP address into its network.

Note



IP addresses can be assigned via BootP under Windows and Linux!

You can use WAGO-BootP-Server to assign an IP address under the Windows and Linux operating systems. You can also use any other BootP server besides WAGO-BootP-Server. You can also use any other BootP server besides the WAGO-BootP-Server.

Information



More information about WAGO-BootP-Server

The process for assigning addresses using WAGO-BootP-Server is described in detail in the section "Commissioning Fieldbus Node".

The BootP Client assists in the dynamic configuration of the network parameters: The ETHERNET TCP/IP fieldbus controller has a BootP client that supports the following options in addition to the default "IP address" option:

Table 74: BootP options

Option	Meaning
[OPT1] Subnet mask	32-bit address mask that displays which bits of the IP address identify the network and which identify the network stations.
[OPT2] Time zone	Time difference between the local time and the UTC (Universal Time Coordinated).
[OPT3] Gateway	IP address of the router that permits access to other networks.
[OPT6] DNS server	IP address of the name servers that converts a name into an IP address. Up to 2 DNS servers can be configured.
[OPT12] Host name	The name of the host is the unique name of a computer in a network. The host name can contain up to 32 characters.
[OPT15] Domain name	The name of the domain is the unique name of a network. The domain name can contain up to 32 characters.
[OPT42] NTP server	IP address of the Network Time Server. When assigning an NTP server, the SNTP client is automatically enabled in the coupler.

The "Features" WBM page can also be used to select the "BootP Request before static IP" option. After the restart, 5 BootP queries are sent. If there is no response to any of these queries, the fieldbus coupler/controller tries to configure itself with the IP parameters saved in the EEPROM.

The network parameters (IP address, etc.) are stored in the EEPROM when using the Bootstrap protocol to configure the node.

Note



BootP configuration is saved in the EEPROM!

Please note that the network configuration is stored in the EEPROM when using BootP in contrast to configuration via DHCP.

By default, BootP is activated in the fieldbus coupler/controller.

When BootP is activated, the fieldbus coupler/controller expects the BootP server to be permanently available.

If there is no BootP server available after a PowerOn reset, the network will remain inactive.

To operate the fieldbus coupler/controller with the IP configuration stored in the EEPROM, you must deactivate the BootP protocol after configuration.

The Web-based management system is used to deactivate the BootP protocol on the respective fieldbus coupler/controller-internal HTML page under the "Port" link.

If BootP is deactivated, the fieldbus coupler/controller uses the parameters saved in the EEPROM when booting next.

If there is an error in the saved parameters, the I/O LED reports a blink code and configuration via BootP is turned on automatically.

12.1.2.2 DHCP (Dynamic Host Configuration Protocol)

The fieldbus coupler/controller internal HTML page opened via the "Port" link provides the option to configure the network using the data saved in the EEPROM or via DHCP instead of via the BootP protocol.

DHCP (Dynamic Host Configuration Protocol) is a further development of BootP and is backwards compatible with BootP.

Both BOOTP and DHCP assign an IP address to the fieldbus node (Client) when starting; the sequence is the same as for BootP.

For configuration of the network parameters via DHCP, the fieldbus coupler/controller sends a client request to the DHCP server e.g., on the connected PC.

A broadcast request is transmitted to Port 67 (DHCP server) via the protocol that contains the hardware address (MAC ID) for the fieldbus coupler/controller.

The DHCP server then receives this message. The server contains a database in which the MAC ID and IP addresses are assigned to one another. When a MAC address is found a broadcast reply is transmitted via the network.

The fieldbus coupler/controller "listens" at the specified Port 68 for a response from the DHCP server. Incoming packets contain information such as the IP address and the MAC address for the fieldbus coupler/controller. A fieldbus coupler/controller recognizes by the MAC address that the message is intended for that particular fieldbus coupler/controller and accepts the transmitted IP address into its network.

If there is no reply, the inquiry is sent again after 4 seconds, 8 seconds and 16 seconds.

If all inquiries receive no reply, a blink code is reported via the I/O LED. The parameters cannot be applied from the EEPROM.

Note



DHCP configuration is not saved in the EEPROM!

Please note that the network configuration is not stored in the EEPROM when using DHCP in contrast to configuration via BootP.

The difference between BOOTP and DHCP is that both use different assignment methods and that configuration with DHCP is time limited. The DHCP client always has to update the configuration after the time has elapsed. Normally, the same parameters are continuously confirmed by the server.

The difference between BOOTP and DHCP is that both use different assignment methods. BOOTP can be used to assign a fixed IP address for each client where the addresses and their reservation are permanently saved in the BOOTP server database.

Because of this time dependency, DHCP is also used to dynamically assign available IP addresses through client leases (lease time after which the client requests a new address) where each DHCP client address is saved temporarily in the server database.

In addition, DHCP clients do not require a system restart to rebind or renew configuration with the DHCP server. Instead, clients automatically enter a rebinding state at set timed intervals to renew their leased address allocation with the DHCP server. This process occurs in the background and is transparent to the user.

There are three different operating modes for a DHCP server:

- **Manual assignment**
In this mode, the IP addresses are permanently assigned on the DHCP server to specific MAC addresses. The addresses are assigned to the MAC address for an indefinite period.
Manual assignments are used primarily to ensure that the DHCP client can be reached under a fixed IP address.
- **Automatic assignment**
For automatic assignment, a range of IP addresses is assigned on the DHCP server.
If the address was assigned from this range once to a DHCP client, then it belongs to the client for an indefinite period as the assigned IP address is also bound to the MAC address.
- **Dynamic assignment**
This process is similar to automatic assignment, but the DHCP server has a statement in its configuration file that specifies how long a certain IP address may be "leased" to a client before the client must log into the server again and request an "extension".
If the client does not log in, the address is released and can be reassigned to another (or the same) client. The time defined by the administrator is called Lease Time.
Some DHCP servers also assign IP addresses based on the MAC address, i.e., a client receives the same IP address as before after longer network absence and elapse of the Lease Time (unless the IP address has been assigned otherwise in the mean time).

DHCP is used to dynamically configure the network parameters.
The ETHERNET TCP/IP fieldbus controller has a DHCP client that supports the following options in addition to the default "IP address" option:

Table 75: Meaning of DHCP options

Option	Meaning
[OPT1] Subnet mask	32-bit address mask that displays which bits of the IP address identify the network and which identify the network stations.
[OPT2] Time zone	Time difference between the local time and the UTC (Universal Time Coordinated).
[OPT3] Gateway	IP address of the router that permits access to other networks.
[OPT6] DNS server	IP address of the name servers that converts a name into an IP address. Up to 2 DNS servers can be configured.
[OPT15] Domain name *)	The name of the domain is the unique name of a network. The domain name can contain up to 32 characters.
[OPT42] NTP server	IP address of the Network Time Server. When assigning an NTP server, the SNTP client is automatically enabled in the coupler.
[OPT51] Lease time	The maximum duration (i.e., how long the fieldbus coupler/controller maintains the assigned IP address) can be defined here. The maximum lease time for the fieldbus controller is 48 days. This is due to the internal timer resolution.
[OPT58] Renewing time	The renewing time indicates when the fieldbus coupler/controller must renew the lease time. The renewing time should be approximately half of the lease time.
[OPT59] Rebinding time	The rebinding time indicates after what amount of time the fieldbus coupler/controller must have received its new address. The rebinding time should be approximately 7/8 of the lease time.

*) In contrast to BootP, the DHCP client does not support assignment of the host name.

12.1.2.3 HTTP (Hypertext Transfer Protocol)

HTTP is a protocol used by WWW (World Wide Web) servers for the forwarding of hypermedia, texts, images, audiodata, etc.

Today, HTTP forms the basis of the Internet and is also based on requests and responses in the same way as the BootP protocol.

The HTTP server implemented in the (programmable) fieldbus coupler or controller is used for viewing the HTML pages saved in the coupler/controller. The HTML pages provide information about the coupler/controller (state, configuration), the network and the process image.

On some HTML pages, (programmable) fieldbus coupler or controller settings can also be defined and altered via the web-based management system (e.g. whether IP configuration of the coupler/controller is to be performed via the DHCP protocol, the BootP protocol or from the data stored in the EEPROM).

The HTTP server uses port number 80.

12.1.2.4 DNS (Domain Name Systems)

The DNS client enables conversion of logical Internet names such as www.wago.com into the appropriate decimal IP address represented with separator stops, via a DNS server. Reverse conversion is also possible.

The addresses of the DNS server are configured via DHCP, BootP or web-based management. Up to 2 DNS servers can be specified. The host identification can be achieved with two functions, an internal host table is not supported.

12.1.2.5 SNTP-Client (Simple Network Time Protocol)

The SNTP client is used for synchronization of the time of day between a time server (NTP and SNTP server Version 3 and 4 are supported) and the clock module integrated in the (programmable) fieldbus coupler or controller. The protocol is executed via a UDP port. Only unicast addressing is supported.

Configuration of the SNTP client

The configuration of the SNTP client is performed via the web-based management system under the “Clock” link. The following parameters must be set:

Table 76: Meaning of the SNTP Parameters

Parameter	Meaning
Adresse des Time-Servers	The address assignment can be made over an IP address.
Time zone	The time zone relative to GMT (Greenwich Mean time). A range of -12 to +12 hours is acceptable.
Update Time	The update time indicates the interval in seconds, in which the synchronization with the time server is to take place.
Enable Time Client	It indicates whether the SNTP Client is to be activated or deactivated.

12.1.2.6 FTP-Server (File Transfer Protocol)

The file transfer protocol (FTP) enables files to be exchanged between different network stations regardless of operating system.

In the case of the ETHERNET coupler/controller, FTP is used to store and read the HTML pages created by the user, the IEC61131 program and the IEC61131 source code in the (programmable) fieldbus coupler or controller.

A total memory of 2 MB is available for the file system.



Note

Cycles for flash limited to 1 million!

Up to 1 million write cycles per sector are allowed when writing the flash for the file system. The file system supports "Wear-Leveling", so that the same sectors are not always written to.



Information

More Information about the implemented Protocols

You can find a list of the exact available implemented protocols in the chapter "Technical Data" to the fieldbus coupler and/or controller.

12.1.2.7 SNMP (Simple Network Management Protocol)

The Simple Network Management Protocol (SNMP) is responsible for transporting the control data that allows the exchange of management information as well as status and statistic data between individual network components and a management system.

An SNMP management workstation polls the SNMP agents to obtain information on the relevant devices.

SNMP is supported in versions 1/2c and some fieldbus couplers/controllers in version 3.

This represents a community message exchange in SNMP versions 1 and 2c. The community name of the network community must thereby be specified.

In SNMP version 3, exchanging messages is user-related. Each device, that knows the passwords set via WBM, may read or write values from the controller. In SNMPv3, user data from SNMP messages can also be transmitted in encoded form. This way, both requested values and values to be written cannot be easily decoded by others via ETHERNET. This is why SNMPv3 is often used in safety-related networks.

The device data, that can be accessed or modified by an SNMP agent, is called SNMP object. The sets of SNMP objects are stored in a logical database called Management Information Base (MIB); this is why these objects are typically known as "MIB objects".

The SNMP of the ETHERNET controller includes both the general MIB acc. to RFC1213 (MIB II) and a special WAGO MIB.

SNMP is processed via port 161. The port number for SNMP traps (agent messages) is 161. Both ports must be enabled to use SNMP.

12.1.2.7.1 MIB II Description

MIB II acc. to RFC1213 is divided into the following groups:

Table 77: MIB II groups

Group	Identifier
System Group	1.3.6.1.2.1.1
Interface Group	1.3.6.1.2.1.2
IP Group	1.3.6.1.2.1.4
IpRoute Table Group	1.3.6.1.2.1.4.21
ICMP Group	1.3.6.1.2.1.5
TCP Group	1.3.6.1.2.1.6
UDP Group	1.3.6.1.2.1.7
SNMP Group	1.3.6.1.2.1.11



Information

Additional Information:

Please find detailed information on these individual groups in section "MIB II groups" of the manual appendix..

12.1.2.7.2 Traps

Standard Traps

For specific events, the SNMP agent will independently send one of the following messages without polling the manager.

Note



Enable event messages (traps) in the WBM!

Initially enable the event messages in the WBM in menu "SNMP" under "Trap Enable". Traps in version 1, 2c and 3 may be activated separately.

The following messages are triggered automatically as traps (SNMPv1) by the fieldbus coupler/controller:

Table 78: Standard Traps

TrapType/TrapNumber/ OID of the provided value	Name	Event
TrapType = 0	ColdStart	Restart the coupler/controller
TrapType = 1	WarmStart	Reset via service switch
TrapType = 3	EthernetUp	Network connection detected
TrapType = 4	AuthenticationFailure	Unauthorized (abortive) MIB access
TrapType = 6/ ab Trap-Nummer 25 benutzerspezifisch	enterpriseSpecific	Enterprise-specific messages and function poll in the PFC program starting with enterprise trap number 25

12.1.3 Application Protocols

If fieldbus specific application protocols are implemented, then the appropriate fieldbus specific communication is possible with the respective coupler/controller. Thus the user is able to have a simple access from the respective fieldbus on the fieldbus node.

The implemented fieldbus specific application protocols these protocols are individual described in the following chapters.

12.2 MODBUS Functions

12.2.1 General

MODBUS is a manufacturer-independent, open fieldbus standard for diverse applications in manufacturing and process automation.

The MODBUS protocol is implemented according to the current Internet Draft of the IETF (Internet Engineering Task Force) and performs the following functions:

- Transmission of the process image
- Transmission of the fieldbus variables
- Transmission of different settings and information on the coupler/controller

The data transmission in the fieldside takes place via TCP and via UDP.

The MODBUS/TCP protocol is a variation of the MODBUS protocol, which was optimized for communication via TCP/IP connections.

This protocol was designed for data exchange in the field level (i.e. for the exchange of I/O data in the process image).

All data packets are sent via a TCP connection with the port number 502.

MODBUS/TCP segment

The general MODBUS/TCP header is as follows:

Table 79: MODBUS/TCP header

Byte	0	1	2	3	4	5	6	7	8...n
	Identifier (entered by receiver)		Protocol- identifier (is always 0)		Length field (High byte, low byte)		Unit identifier (Slave address)	MODBUS function code	Data



Information

Additional Information

The structure of a datagram is specific for the individual function. Refer to the descriptions of the MODBUS Function codes.

For the MODBUS protocol 15 connections are made available over TCP. Thus it allows digital and analog output data to be directly read out at a fieldbus node and special functions to be executed by way of simple MODBUS function codes from 15 stations simultaneously.

For this purpose a set of MODBUS functions from the Open MODBUS/TCP specification is realized.

Information



More information

More information on the “Open MODBUS/TCP specification” you can find in the Internet: www.modbus.org.

Therefore the MODBUS protocol based essentially on the following basic data types:

Table 80: Basic data types of MODBUS protocol

Data type	Length	Description
Discrete Inputs	1 Bit	Digital inputs
Coils	1 Bit	Digital outputs
Input Register	16 Bit	Analog input data
Holding Register	16 Bit	Analog output data

For each basic data type one or more function codes are defined.

These functions allow digital or analog input and output data, and internal variables to be set or directly read out of the fieldbus node.

Table 81: List of the MODBUS functions in the fieldbus controller

Function code	Function	Access method and description	Access to resources
FC1 0x01	Read Coils	Reading of several single input bits	R: Process image, PFC variables
FC2 0x02	Read Input Discretes	Reading of several input bits	R: Process image, PFC variables
FC3 0x03	Read Multiple Registers	Reading of several input registers	R: Process image, PFC variables, internal variables, NOVRAM
FC4 0x04	Read Input Registers	Reading of several input registers	R: Process image, PFC variables, internal variables, NOVRAM
FC5 0x05	Write Coil	Writing of an individual output bit	W: Process image, PFC variables
FC6 0x06	Write Single Register	Writing of an individual output register	W: Process image, PFC variables, internal variables, NOVRAM
FC11 0x0B	Get Comm Event Counters	Communication event counter	R: None
FC15 0x0F	Force Multiple Coils	Writing of several output bits	W: Process image, PFC variables
FC16 0x10	Write Multiple Registers	Writing of several output registers	W: Process image, PFC variables, internal variables, NOVRAM
FC22 0x16	Mask Write Register		W: Process image, PFC variables, NOVRAM
FC23 0x17	Read/Write Registers	Reading and writing of several output registers	R/W: Process image, PFC variables, NOVRAM

To execute a desired function, specify the respective function code and the address of the selected input or output data.

Note



Note the number system when addressing!

The examples listed use the hexadecimal system (i.e.: 0x000) as their numerical format. Addressing begins with 0. The format and beginning of the addressing may vary according to the software and the control system. All addresses then need to be converted accordingly.

12.2.2 Use of the MODBUS Functions

The example below uses a graphical view of a fieldbus node to show which MODBUS functions can be used to access data of the process image.

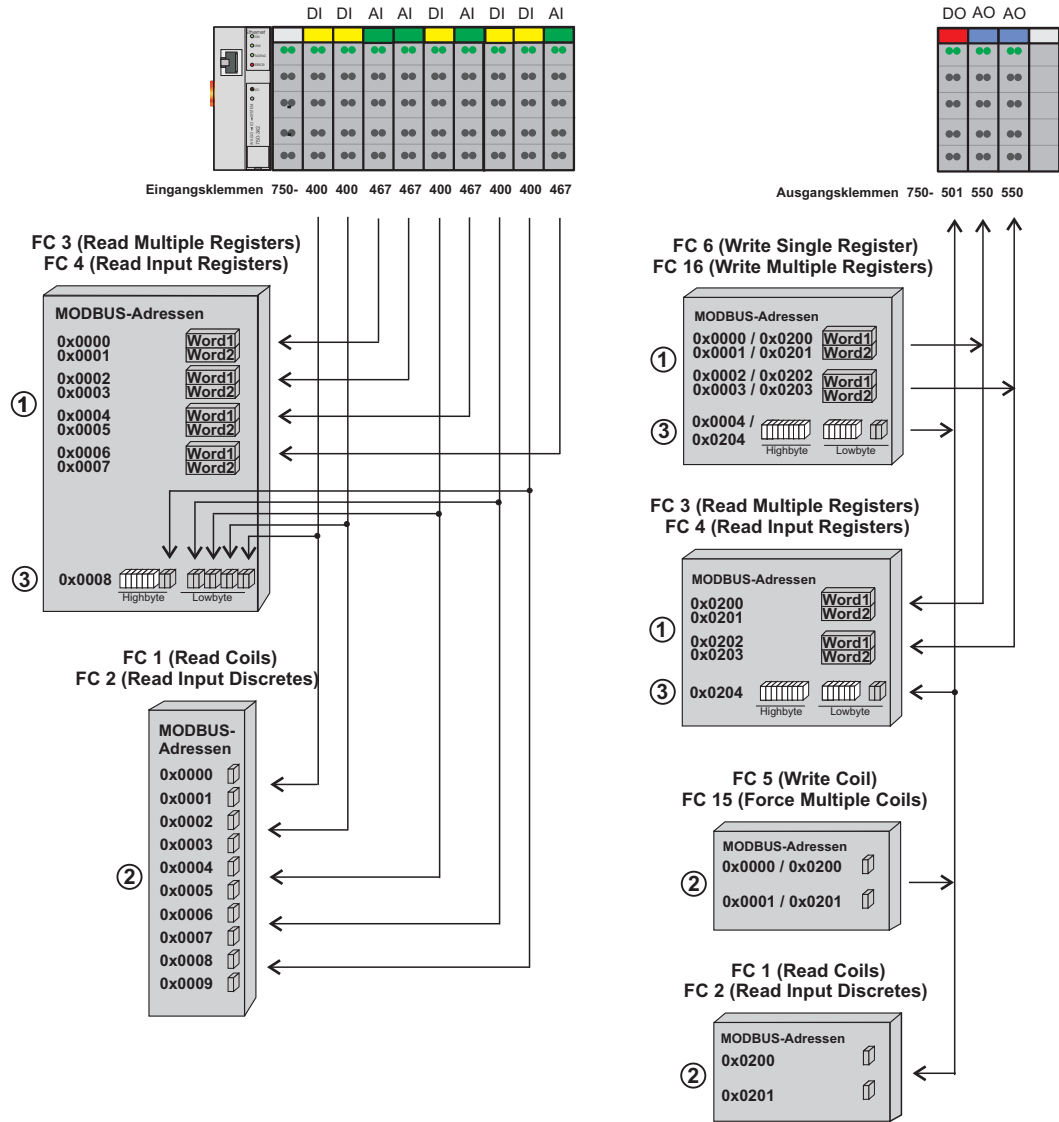


Figure 76: Use of the MODBUS Functions



Note

Use register functions to access analog signals and coil functions to access binary signals!

It is recommended that analog data be accessed with register functions ① and digital data with coil functions ②. If reading or writing access to binary signals is performed via register functions ③, an address shift may occur as soon as further analog modules are operated on the coupler/controller.

12.2.3 Description of the MODBUS Functions

All MODBUS functions are executed as follows:

1. A MODBUS TCP master (e.g., a PC) makes a request to the WAGO fieldbus node using a specific function code based on the desired operation..
2. The WAGO fieldbus node receives the datagram and then responds to the master with the proper data, which is based on the master's request.

If the WAGO fieldbus node receives an incorrect request, it sends an error datagram (Exception) to the master.

The exception code contained in the exception has the following meaning:

Table 82: Exception Codes

Exception Code	Meaning
0x01	Illegal function
0x02	Illegal data address
0x03	Illegal data value
0x04	Slave device failure

The following chapters describe the datagram architecture of request, response and exception with examples for each function code.



Note

Reading and writing of outputs via FC1 to FC4 is also possible by adding an offset!

In the case of the read functions (FC1 ... FC4) the outputs can be additionally written and read back by adding an offset of 200hex (0x0200) to the MODBUS addresses in the range of [0_{hex} ... FF_{hex}] and an offset of 1000_{hex} (0x01000) to the MODBUS addresses in the range of [6000_{hex} ... 62FC_{hex}].

12.2.3.1 Function Code FC1 (Read Coils)

This function reads the status of the input and output bits (coils) in a slave device.

Request

The request specifies the reference number (starting address) and the bit count to read.

Example: Read output bits 0 to 7.

Table 83: Request of Function code FC1

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x01
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0008

Response

The current values of the response bits are packed in the data field. A binary 1 corresponds to the ON status and a 0 to the OFF status. The lowest value bit of the first data byte contains the first bit of the request. The others follow in ascending order. If the number of inputs is not a multiple of 8, the remaining bits of the last data byte are filled with zeroes (truncated).

Table 84: Response of Function code FC1

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x01
Byte 8	Byte count	0x01
Byte 9	Bit values	0x12

The status of the inputs 7 to 0 is shown as byte value 0x12 or binary 0001 0010. Input 7 is the bit having the highest significance of this byte and input 0 the lowest value.

The assignment is thus made from 7 to 0 as follows:

Table 85: Assignment of inputs

	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF
Bit	0	0	0	1	0	0	1	0
Coil	7	6	5	4	3	2	1	0

Exception

Table 86: Exception of Function code FC1

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x81
Byte 8	Exception code	0x01 or 0x02

12.2.3.2 Function Code FC2 (Read Input Discretes)

This function reads the input bits from a slave device.

Request

The request specifies the reference number (starting address) and the bit count to be read.

Example: Read input bits 0 to 7

Table 87: Request of Function code FC2

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x02
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0008

Response

The current value of the requested bits are packed into the data field. A binary 1 corresponds to the ON status and a 0 the OFF status. The lowest value bit of the first data byte contains the first bit of the inquiry. The others follow in an ascending order. If the number of inputs is not a multiple of 8, the remaining bits of the last data byte are filled with zeroes (truncated).

Table 88: Response of Function code FC2

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x02
Byte 8	Byte count	0x01
Byte 9	Bit values	0x12

The status of the inputs 7 to 0 is shown as a byte value 0x12 or binary 0001 0010. Input 7 is the bit having the highest significance of this byte and input 0 the lowest value. The assignment is thus made from 7 to 0 as follows:

Table 89: Assignment of inputs

	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF
Bit	0	0	0	1	0	0	1	0
Coil	7	6	5	4	3	2	1	0

Exception

Table 90: Exception of Function code FC2

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x82
Byte 8	Exception code	0x01 or 0x02

12.2.3.3 Function Code FC3 (Read Multiple Registers)

This function reads the contents of holding registers from a slave device in word format.

Request

The request specifies the reference number (start register) and the word count (register quantity) of the registers to be read. The reference number of the request is zero based, therefore, the first register starts at address 0.

Example: Read registers 0 and 1.

Table 91: Request of Function code FC3

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x03
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002

Response

The reply register data is packed as 2 bytes per register. The first byte contains the higher value bits, the second the lower values.

Table 92: Response of Function code FC3

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x03
Byte 8	Byte count	0x04
Byte 9, 10	Value register 0	0x1234
Byte 11, 12	Value register 1	0x2345

The contents of register 0 are displayed by the value 0x1234 and the contents of register 1 is 0x2345.

Exception

Table 93: Exception of Function code FC3

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x83
Byte 8	Exception code	0x01 or 0x02

12.2.3.4 Function Code FC4 (Read Input Registers)

This function reads contents of input registers from the slave device in word format.

Request

The request specifies a reference number (start register) and the word count (register quantity) of the registers to be read. The reference number of the request is zero based, therefore, the first register starts at address 0.

Example: Read registers 0 and 1

Table 94: Request of Function code FC4

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x04
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002

Response

The register data of the response is packed as 2 bytes per register. The first byte has the higher value bits, the second the lower values.

Table 95: Response of Function code FC4

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x04
Byte 8	Byte count	0x04
Byte 9, 10	Value register 0	0x1234
Byte 11, 12	Value register 1	0x2345

The contents of register 0 are shown by the value 0x1234 and the contents of register 1 is 0x2345.

Exception

Table 96: Exception of Function code FC4

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x84
Byte 8	Exception code	0x01 or 0x02

12.2.3.5 Function Code FC5 (Write Coil)

This function writes a single output bit to the slave device.

Request

The request specifies the reference number (output address) of output bit to be written. The reference number of the request is zero based; therefore, the first coil starts at address 0.

Example: Turn ON the second output bit (address 1)

Table 97: Request of Function code FC5

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x05
Byte 8, 9	Reference number	0x0001
Byte 10	ON/OFF	0xFF
Byte 11		0x00

Response

Table 98: Response of Function code FC5

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x05
Byte 8, 9	Reference number	0x0001
Byte 10	Value	0xFF
Byte 11		0x00

Exception

Table 99: Exception of Function code FC5

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01, 0x02 or 0x03

12.2.3.6 Function Code FC6 (Write Single Register)

This function writes the value of one single output register to a slave device in word format.

Request

The request specifies the reference number (register address) of the first output word to be written. The value to be written is specified in the “Register Value” field. The reference number of the request is zero based; therefore, the first register starts at address 0.

Example: Write a value of 0x1234 to the second output register

Table 100: Request of Function code FC6

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x06
Byte 8, 9	Reference number	0x0001
Byte 10, 11	Register value	0x1234

Response

The reply is an echo of the inquiry.

Table 101: Response of Function code FC6

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x06
Byte 8, 9	Reference number	0x0001
Byte 10, 11	Register value	0x1234

Exception

Table 102: Exception of Function code FC6

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.7 Function Code FC11 (Get Comm Event Counter)

This function returns a status word and an event counter from the slave device's communication event counter. By reading the current count before and after a series of messages, a master can determine whether the messages were handled normally by the slave.

Following each successful new processing, the counter counts up. This counting process is not performed in the case of exception replies, poll commands or counter inquiries.

Request

Table 103: Request of Function code FC11

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0002
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x0B

Response

The reply contains a 2-byte status word and a 2-byte event counter. The status word only contains zeroes.

Table 104: Response of Function code FC11

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x0B
Byte 8, 9	Status	0x0000
Byte 10, 11	Event count	0x0003

The event counter shows that 3 (0x0003) events were counted.

Exception

Table 105: Exception of Function code FC 11

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.8 Function Code FC15 (Force Multiple Coils)

This function sets a sequence of output bits to 1 or 0 in a slave device. The maximum number is 256 bits.

Request

The request message specifies the reference number (first coil in the sequence), the bit count (number of bits to be written), and the output data. The output coils are zero-based; therefore, the first output point is 0.

In this example 16 bits are set, starting with the address 0. The request contains 2 bytes with the value 0xA5F0, or 1010 0101 1111 0000 in binary format.

The first data byte transmits the value of 0xA5 to the addresses 7 to 0, whereby 0 is the lowest value bit. The next byte transmits 0xF0 to the addresses 15 to 8, whereby the lowest value bit is 8.

Table 106: Request of Function code FC15

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0009
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x0F
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0010
Byte 12	Byte count	0x02
Byte 13	Data byte1	0xA5
Byte 14	Data byte2	0xF0

Response

Table 107: Response of Function code FC15

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x0F
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0010

Exception

Table 108: Exception of Function code FC15

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x8F
Byte 8	Exception code	0x01 or 0x02

12.2.3.9 Function Code FC16 (Write Multiple Registers)

This function writes a sequence of registers in a slave device in word format.

Request

The Request specifies the reference number (starting register), the word count (number of registers to write), and the register data. The data is sent as 2 bytes per register. The registers are zero-based; therefore, the first output is at address 0. Example: Set data in registers 0 and 1

Table 109: Request of Function code FC16

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x000B
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002
Byte 12	Byte count	0x04
Byte 13, 14	Register value 1	0x1234
Byte 15, 16	Register value 2	0x2345

Response

Table 110: Response of Function code FC16

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002

Exception

Table 111: Exception of Function code FC16

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.10 Function Code FC22 (Mask Write Register)

This function manipulates individual bits within a register using a combination of an AND mask, an OR mask, and the register's current content.

Request

Table 112: Request of Function code FC22

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0002
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x16
Byte 8, 9	Reference number	0x0000
Byte 10, 11	AND mask	0x0000
Byte 12, 13	OR mask	0xAAAA

Response

Table 113: Response of Function code FC22

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	AND mask	0x0000
Byte 12, 13	OR mask	0xAAAA

Exception

Table 114: Exception of Function code FC22

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.11 Function Code FC23 (Read/Write Multiple Registers)

This function performs a combination of a read and write operation in a single request. The function can write the new data to a group registers, and then return the data of a different group.

Request

The reference numbers (addresses) are zero-based in the request message; therefore, the first register is at address 0.

The request message specifies the registers to read and write. The data is sent as 2 bytes per register.

Example: The data in register 3 is set to value 0x0123, and values 0x0004 and 0x5678 are read out of the two registers 0 and 1.

Table 115: Request of Function code FC23

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x000F
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x17
Byte 8, 9	Reference number for read	0x0000
Byte 10, 11	Word count for read (1...125)	0x0002
Byte 12, 13	Reference number for write	0x0003
Byte 14, 15	Word count for write (1...100)	0x0001
Byte 16	Byte count (2 x word count for write)	0x02
Byte 17...(B+16)	Register values (B = Byte count)	0x0123

Response

Table 116: Response of Function code FC23

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x17
Byte 8	Byte count (2 x word count for read)	0x04
Byte 9...(B+8)	Register values (B = Byte count)	0x0004 or 0x5678

Exception

Table 117: Exception of Function code FC23

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x97
Byte 8	Exception code	0x01 or 0x02



Note

Note that if the register ranges overlap, the results are undefined!
If register areas for read and write overlap, the results are undefined.

12.2.4 MODBUS Register Mapping

The following tables display the MODBUS addressing and the corresponding IEC61131 addressing for the process image, the PFC variables, the NOVDRAM data, and the internal variables is represented.

Via the register services the states of the complex and digital I/O modules can be determined or changed.

Register Access Reading (with FC3, FC4 and FC23)

Table 118: Register access reading (with FC3, FC4 and FC23)

MODBUS address		IEC 61131	Memory range
[dec]	[hex]	address	
0...255	0x0000...0x00FF	%IW0...%IW255	Physical input area (1) First 256 words of physical input data
256...511	0x0100...0x01FF	%QW256...%QW511	PFC OUT area Volatile PFC output variables
512...767	0x0200...0x02FF	%QW0...%QW255	Physical output area (1) First 256 words of physical output data
768...1023	0x0300...0x03FF	%IW256...%IW511	PFC IN area Volatile PFC input variables
1024...4095	0x0400...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...12287	0x1000...0x2FFF	-	Configuration register (see following chapter "Configuration Functions")
12288...24575	0x3000...0x5FFF	%MW0...%MW12287	NOVDRAM 24 kB retain memory *) *) In Target settings RETAIN on 0, flags on MAX (24 kB)
24576...25340	0x6000...0x62FC	%IW512...%IW1275	Physical input area (2) Additional 764 words physical input data
25341...28671	0x62FD...0x6FFF	-	MODBUS exception: "Illegal data address"
28672...29435	0x7000...0x72FB	%QW512...%QW1275	Physical output area (2) Additional 764 words physical output data
29436...32767	0x72FC...0x7FFF	-	MODBUS exception: "Illegal data address"
32768...36863	0x8000...0x8FFF	-	NOVDRAM Additional, 16-bit length
36864...65535	0x9000...0xFFFF	-	MODBUS exception: "Illegal data address"

Register Access Writing (with FC6, FC16, FC22 and FC23)

Table 119: Register access writing (with FC6, FC16, FC22 and FC23)

MODBUS address		IEC 61131 address	Memory range
[dec]	[hex]		
0...255	0x0000...0x00FF	%QW0...%QW255	Physical output area (1) First 256 words of physical output data
256...511	0x0100...0x01FF	%IW256...%IW511	PFC IN area Volatile PFC input variables
512...767	0x0200...0x02FF	%QW0...%QW255	Physical output area (1) First 256 words of physical output data
768...1023	0x0300...0x03FF	%IW256...%IW511	PFC IN area Volatile PFC input variables
1024...4095	0x0400...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...12287	0x1000...0x2FFF	-	Configuration register (see following chapter „Configuration Functions“)
12288...24575	0x3000...0x5FFF	%MW0...%MW12287	NOVRAM 24 kB retain memory *) *) In Target settings RETAIN on 0, flags on MAX (24 kB)
24576...25340	0x6000...0x62FC	%QW512...%QW1275	Physical output area (2) Additional 764 words physical output data
25341...28671	0x62FD...0x6FFF	-	MODBUS exception: "Illegal data address"
28672...29435	0x7000...0x72FB	%QW512...%QW1275	Physical output area (2) Additional 764 words physical output data
29436...32767	0x72FC...0x7FFF	-	MODBUS exception: "Illegal data address"
32768...36863	0x8000...0x8FFF	-	NOVRAM Additional, 16-bit length
36864...65535	0x9000...0xFFFF	-	MODBUS exception: "Illegal data address"

The digital MODBUS services (coil services) are bit accesses, with which only the states of digital I/O modules can be determined or changed. Complex I/O modules are not attainable with these services and so they are ignored. Because of this the addressing of the digital channels begins again with 0, so that the MODBUS address is always identical to the channel number, (i.e. the digital input no. 47 has the MODBUS address "46").

Bit Access Reading (with FC1 and FC2)

Table 120: Bit access reading (with FC1 and FC2)

MODBUS address		Memory range	Description
[dec]	[hex]		
0...511	0x0000...0x01FF	Physical input area (1)	First 512 digital inputs
512...1023	0x0200...0x03FF	Physical output area (1)	First 512 digital outputs
1024...4095	0x0400...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...8191	0x1000...0x1FFF	%QX256.0...%QX511.15	PFC OUT area Volatile PFC output variables
8192...12287	0x2000...0x2FFF	%IX256.0...%IX511.15	PFC IN area Volatile PFC input variables
12288...32767	0x3000...0x7FFF	%MX0...%MX1279.15	NOVRAM 2 kB retain memory (max. 24 kB)
32768...34295	0x8000...0x85F7	Physical input area (2)	Starts with the 513 th and ends with the 2039 th digital input
34296...36863	0x85F8...0x8FFF	-	MODBUS exception: "Illegal data address"
36864...38391	0x9000...0x95F7	Physical output area (2)	Starts with the 513 th and ends with the 2039 th digital output
38392...65535	0x95F8...0xFFFF	-	MODBUS exception: "Illegal data address"

Bit Access Writing (with FC5 and FC15)

Table 121: Bit access writing (with FC5 and FC15)

MODBUS address		Memory Range	Description
[dez]	[hex]		
0...511	0x0000...0x01FF	Physical output area (1)	First 512 digital outputs
512...1023	0x0200...0x03FF	Physical output area (1)	First 512 digital outputs
1024...4095	0x0400...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...8191	0x1000...0x1FFF	%IX256.0...%IX511.15	PFC IN area Volatile PFC input variables
8192...12287	0x2000...0x2FFF	%IX256.0...%IX511.15	PFC IN area Volatile PFC input variables
12288...32767	0x3000...0x7FFF	%MX0...%MX1279.15	NOVRAM 2 kB retain memory
32768...34295	0x8000...0x85F7	Physical output area (2)	Starts with the 513 th and ends with the 2039 th digital input
34296...36863	0x85F8...0x8FFF	-	MODBUS-Exception: "Illegal data address"
36864...38391	0x9000...0x95F7	Physical output area (2)	Starts with the 513 th and ends with the 2039 th digital output
38392...65535	0x95F8...0xFFFF	-	MODBUS-Exception: "Illegal data address"

12.2.5 MODBUS Registers

Table 122: MODBUS registers

Register address	Access	Length (Word)	Description
0x1000	R/W	1	Watchdog time read/write
0x1001	R/W	1	Watchdog coding mask 1...16
0x1002	R/W	1	Watchdog coding mask 17...32
0x1003	R/W	1	Watchdog trigger
0x1004	R	1	Minimum trigger time
0x1005	R/W	1	Watchdog stop (Write sequence 0xAAAA, 0x5555)
0x1006	R	1	Watchdog status
0x1007	R/W	1	Restart watchdog (Write sequence 0x1)
0x1008	R/W	1	Stop watchdog (Write sequence 0x55AA or 0xAA55)
0x1009	R/W	1	MODBUS and HTTP close at watchdog time-out
0x100A	R/W	1	Watchdog configuration
0x100B	W	1	Save watchdog parameter
0x1020	R	1...2	LED error code
0x1021	R	1	LED error argument
0x1022	R	1...4	Number of analog output data in the process image (in bits)
0x1023	R	1...3	Number of analog input data in the process image (in bits)
0x1024	R	1...2	Number of digital output data in the process image (in bits)
0x1025	R	1...4	Number of digital input data in the process image (in bits)
0x1028	R/W	1	Boot configuration
0x1029	R	9	MODBUS/TCP statistics
0x102A	R	1	Number of TCP connections
0x102B	W	1	KBUS Reset
0x1030	R/W	1	Configuration MODBUS/TCP time-out
0x1031	R	3	Read out the MAC-ID of the coupler/controller
0x1035	R/W	1	Timeoffset RTC
0x1036	R/W	1	Daylight Saving
0x1037	R/W	1	Modbus Response Delay (ms)
0x1050	R	3	Diagnosis of the connected I/O modules
0x2000	R	1	Constant 0x0000
0x2001	R	1	Constant 0xFFFF
0x2002	R	1	Constant 0x1234
0x2003	R	1	Constant 0xAAAA
0x2004	R	1	Constant 0x5555
0x2005	R	1	Constant 0x7FFF
0x2006	R	1	Constant 0x8000
0x2007	R	1	Constant 0x3FFF
0x2008	R	1	Constant 0x4000
0x2010	R	1	Firmware version
0x2011	R	1	Series code
0x2012	R	1	Coupler/controller code
0x2013	R	1	Firmware version major revision
0x2014	R	1	Firmware version minor revision

Table 123: MODBUS registers (Continuation)

Register address	Access	Length (Word)	Description
0x2020	R	16	Short description controller
0x2021	R	8	Compile time of the firmware
0x2022	R	8	Compile date of the firmware
0x2023	R	32	Indication of the firmware loader
0x2030	R	65	Description of the connected I/O modules (module 0...64)
0x2031	R	64	Description of the connected I/O modules (module 65...128)
0x2032	R	64	Description of the connected I/O modules (module 129...192)
0x2033	R	63	Description of the connected I/O modules (module 193...255)
0x2040	W	1	Software reset (Write sequence 0x55AA or 0xAA55)
0x2041	W	1	Format flash disk
0x2042	W	1	Extract HTML sides from the firmware
0x2043	W	1	Factory settings

12.2.5.1 Accessing Register Values

You can use any MODBUS application to access (read from or write to) register values. Both commercial (e.g., "Modscan") and free programs (from <http://www.modbus.org/tech.php>) are available.

The following sections describe how to access both the registers and their values.

12.2.5.2 Watchdog Registers

The watchdog monitors the data transfer between the fieldbus master and the controller. Every time the controller receives a specific request (as define in the watchdog setup registers) from the master, the watchdog timer in the controller resets.

In the case of fault free communication, the watchdog timer does not reach its end value. After each successful data transfer, the timer is reset.

If the watchdog times out, a fieldbus failure has occurred. In this case, the fieldbus controller answers all following MODBUS TCP/IP requests with the exception code 0x0004 (Slave Device Failure).

In the controller special registers are used to setup the watchdog by the master (Register addresses 0x1000 to 0x1008).

By default, the watchdog is not enabled when you turn the controller on. To activate it, the first step is to set/verify the desired time-out value of the Watchdog Time register (0x1000). Second, the function code mask must be specified in the mask register (0x1001), which defines the function code(s) that will reset the timer for the first time. Finally, the Watchdog-Trigger register (0x1003) or the register 0x1007 must be changed to a non-zero value to start the timer subsequently.

Reading the Minimum Trigger time (Register 0x1004) reveals whether a watchdog fault occurred. If this time value is 0, a fieldbus failure is assumed. The timer of watchdog can manually be reset, if it is not timed out, by writing a value of 0x1 to the register 0x1003 or to the Restart Watchdog register 0x1007.

After the watchdog is started, it can be stopped by the user via the Watchdog Stop register (0x1005) or the Simply Stop Watchdog register (0x1008).

The watchdog registers can be addressed in the same way as described with the MODBUS read and write function codes. Specify the respective register address in place of the reference number.

Table 124: Register address 0x1000

Register address 0x1000 (4096_{dec})	
Value	Watchdog time, WS_TIME
Access	Read/write
Default	0x0064
Description	This register stores the watchdog timeout value as an unsigned 16 bit value. The default value is 0. Setting this value will not trigger the watchdog. However, a non zero value must be stored in this register before the watchdog can be triggered. The time value is stored in multiples of 100ms (e.g., 0x0009 is .9 seconds). It is not possible to modify this value while the watchdog is running.

Table 125: Register address 0x1001

Register address 0x1001 (4097_{dec})	
Value	Watchdog function coding mask, function code 1...16, WDFCM_1_16
Access	Read/write
Default	0xFFFF
Description	Using this mask, the function codes can be set to trigger the watchdog function. The function code can be selected via a "1" FC 1 Bit 0 FC 2 Bit 1 FC 3 Bit 2 FC 4 Bit 3 FC 5 Bit 4 ... FC 16 Bit 15 Changes to the register value can only be made if the watchdog is deactivated. The bit pattern stored in the register defines the function codes that trigger the watchdog. Some function codes are not supported. For those the watchdog will not be triggered even if another MODBUS device transmits one of them.

Table 126: Register address 0x1002

Register address 0x1002 (4098 _{dec})	
Value	Watchdog function coding mask, function code 17...32, WD_FCM_17_32
Access	Read/write
Default	0xFFFF
Description	<p>Same function as above, however, with the function codes 17 to 32.</p> <p>FC 17 Bit 0 FC 18 Bit 1 ... FC 32 Bit 15</p> <p>These codes are currently not supported, for this reason the default value should not be changed. Changes to the register value can only be made if the watchdog is deactivated. It is not possible to modify this value while the watchdog is running.</p>

Table 127: Register address 0x1003

Register address 0x1003 (4099 _{dez})	
Value	Watchdog Trigger, WD_TRIGGER
Access	Read/write
Standard	0x0000
Description	This register is used for an alternative trigger method. The watchdog is triggered by writing different values in this register. Values following each other must differ in size. Writing of a value not equal to zero starts the watchdog after a Power-on. For a restart the written value must necessarily be unequal the before written value! A watchdog fault is reset and writing process data is possible again.

Table 128: Register address 0x1004

Register address 0x1004 (4100 _{dez})	
Value	Minimum current trigger time, WD_AC_TRG_TIME
Access	Read/write
Standard	0xFFFF
Description	This register saves the minimum current watchdog trigger time. If the watchdog is triggered, the saved value is compared with the current value. If the current value is smaller than the saved value, this is replaced by the current value. The unit is 100 ms/digit. The saved value is changed by writing new values, which does not affect the watchdog. 0x000 is not permissible.

Table 129: Register address 0x1005

Register address 0x1005 (4101 _{dez})	
Value	Stop watchdog, WD_AC_STOP_MASK
Access	Read/write
Standard	0x0000
Description	The watchdog is stopped if here the value 0xAAAA is written first, followed by 0x5555. The watchdog fault reaction is blocked. A watchdog fault is reset and writing on the process data is possible again.

Table 130: Register address 0x1006

Register address 0x1006 (4102 _{dez})	
Value	While watchdog is running, WD_RUNNING
Access	Read
Standard	0x0000
Description	Current watchdog status. at 0x0000: Watchdog not active at 0x0001: Watchdog active at 0x0002: Watchdog exhausted.

Table 131: Register address 0x1007

Register address 0x1007 (4103 _{dez})	
Value	Restart watchdog, WD_RESTART
Access	Read/write
Standard	0x0001
Description	This register restarts the watchdog timer by writing a value of 0x1 into it. If the watchdog was stopped before the overrun, it is not restarted.

Table 132: Register address 0x1008

Register address 0x1008 (4104 _{dez})	
Value	Simply stop watchdog, WD_AC_STOP_SIMPLE
Access	Read/write
Standard	0x0000
Description	This register stops the watchdog by writing the value 0x0AA55 or 0X55AA into it. The watchdog timeout fault is deactivated and it is possible to write in the watchdog register again. If there is an existing watchdog fault, it is reset

Table 133: Register address 0x1009

Register address 0x1009 (4105 _{dez})	
Value	Close MODBUS socket after watchdog timeout
Access	Read/write
Description	0: MODBUS socket is not closed 1: MODBUS socket is closed

Table 134: Register address 0x100A

Register address 0x100A (4106 _{dez})	
Value	Alternative watchdog
Access	Read/write
Standard	0x0000
Description	This register provides an alternate way to activate the watchdog timer. Procedure: Write a time value in register 0x1000; then write a 0x0001 into register 0x100A. With the first MODBUS request, the watchdog is started. The watchdog timer is reset with each MODBUS/TCP instruction. If the watchdog times out, all outputs are set to zero. The outputs will become operational again, after communications are re-established. Register 0x00A is non-volatile, including register 0x1000. It is not possible to modify the time value in register 0x1000 while the watchdog is running.

The length of each register is 1 word; i.e., with each access only one word can be written or read. Following are two examples of how to set the value for a time overrun:

Setting the watchdog for a timeout of more than 1 second:

1. Write 0x000A in the register for time overrun (0x1000).
Register 0x1000 works with a multiple of 100 ms;
1 s = 1000 ms; 1000 ms / 100 ms = 10_{dec} = A_{hex})
2. Use the function code 5 to write 0x0010 (=2(5-1)) in the coding mask (register 0x1001).

Table 135: Starting Watchdog

FC	FC16	FC15	FC14	FC13	FC12	FC11	FC10	FC9	FC8	FC7	FC6	FC5	FC4	FC3	FC2	FC1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bin	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
hex	0				0				1				0			

Function code 5 (writing a digital output bit) continuously triggers the watchdog to restart the watchdog timer again and again within the specified time. If time between requests exceeds 1 second, a watchdog timeout error occurs.

3. To stop the watchdog, write the value 0x0AA55 or 0X55AA into 0x1008 (Simply Stop Watchdog register, WD_AC_STOP_SIMPLE).

Setting the watchdog for a timeout of 10 minutes or more:

1. Write 0x1770 (= 10*60*1000 ms / 100 ms) in the register for time overrun (0x1000).
(Register 0x1000 works with a multiple of 100 ms;
10 min = 600,000 ms; 600,000 ms / 100 ms = 6000_{dec} = 1770_{hex})
2. Write 0x0001 in the watchdog trigger register (0x1003) to start the watchdog.
3. Write different values (e.g., counter values 0x0000, 0x0001) in the watchdog to trigger register (0x1003).

Values following each other must differ in size. Writing of a value not equal to zero starts the watchdog. Watchdog faults are reset and writing process data is possible again.

4. To stop the watchdog, write the value 0x0AA55 or 0X55AA into 0x1008 (Simply Stop Watchdog register, WD_AC_STOP_SIMPLE).

Table 136: Register address 0x100B

Register address 0x100B (4107_{dez})	
Value	Save watchdog parameter
Access	Write
Standard	0x0000
Description	With writing of '0x55AA' or '0xAA55' in register 0x100B the registers 0x1000, 0x1001, 0x1002 are set on remanent.

12.2.5.3 Diagnostic Registers

The following registers can be read to determine errors in the node:

Table 137: Register address 0x1020

Register address 0x1020 (4128 _{dec})	
Value	LedErrCode
Access	Read
Description	Declaration of the Error code

Table 138: Register address 0x1021

Register address 0x1021 (4129 _{dec})	
Value	LedErrArg
Access	Read
Description	Declaration of the Error argument

12.2.5.4 Configuration Registers

The following registers contain configuration information of the connected modules:

Table 139: Register address 0x1022

Register address 0x1022 (4130 _{dec})	
Value	CnfLen.AnalogOut
Access	Read
Description	Number of word-based outputs registers in the process image in bits (divide by 16 to get the total number of analog words)

Table 140: Register address 0x1023

Register address 0x1023 (4131 _{dec})	
Value	CnfLen.AnalogInp
Access	Read
Description	Number of word-based inputs registers in the process image in bits (divide by 16 to get the total number of analog words)

Table 141: Register address 0x1024

Register address 0x1024 (4132 _{dec})	
Value	CnfLen.DigitalOut
Access	Read
Description	Number of digital output bits in the process image

Table 142: Register address 0x1025

Register address 0x1025 (4133 _{dec})	
Value	CnfLen.DigitalInp
Access	Read
Description	Number of digital input bits in the process image

Table 143: Register address 0x1028

Register address 0x1028 (4136 _{dec})	
Value	Boot options
Access	Read/write
Description	Boot configuration: 1: BootP 2: DHCP 4: EEPROM

Table 144: Register address 0x1029

Register address 0x1029 (4137 _{dec}) with 9 words	
Value	MODBUS TCP statistics
Access	Read/write
Description	<p>1 word SlaveDeviceFailure → internal bus error, fieldbus error by activated watchdog</p> <p>1 word BadProtocol → error in the MODBUS TCP header</p> <p>1 word BadLength → Wrong telegram length</p> <p>1 word BadFunction → Invalid function code</p> <p>1 word BadAddress → Invalid register address</p> <p>1 word BadData → Invalid value</p> <p>1 word TooManyRegisters → Number of the registers which can be worked on is too large, Read/Write 125/100</p> <p>1 word TooManyBits → Number of the coils which can be worked on is too large, Read/Write 2000/800</p> <p>1 word ModTcpMessageCounter → Number of received MODBUS/TCP requests</p> <p>With Writing 0xAA55 or 0x55AA in the register will reset this data area.</p>

Table 145: Register address 0x102A

Register address 0x102A (4138 _{dec}) with a word count of 1	
Value	MODBUS/TCP Connections
Access	Read
Description	Number of TCP connections

Table 146: Register address 0x102B

Register address 0x102B (4139 _{dec}) with a word count of up to 1	
Value	KBUS reset
Access	Write
Description	Writing of this register restarts the internal bus

Table 147: Register address 0x1030

Register address 0x1030 (4144 _{dec}) with a word count of 1	
Value	Configuration MODBUS/TCP Time-out
Access	Read/write
Default	0x0258 (600 decimal)
Description	This is the maximum number of milliseconds the fieldbus coupler will allow a MODBUS/TCP connection to stay open without receiving a MODBUS request. Upon time-out, idle connection will be closed. Outputs remain in last state. Default value is 600 ms (60 seconds), the time base is 100 ms, the minimal value is 100 ms. If the value is set to '0', the timeout is disabled. On this connection, the watchdog is triggered with a request.

Table 148: Register address 0x1031

Register address 0x1031 (4145 _{dec}) with a word count of 3	
Value	Read the MAC-ID of the controller
Access	Read
Description	This register gives the MAC-ID, with a length of 3 words

Table 1: Register address 0x1035

Register address 0x1035 (4149_{dez}) 1 Word	
Value	Configuration of the time offsets to the GMT time
Access	Read/write
Default	0x0000
Description	Register to set the time offset to the UTC time (Greenwich meridian) with a possible setting range from -12 to +12.

Table 1: Register address 0x1036

Register address 0x1036 (4150_{dez}) 1 Word	
Value	Configuration of summer or winter time
Access	Read/write
Default	0x0000
Description	Register to set winter or summer time (Daylight Saving Time). The values 0 and 1 are valid.

Table 1: Register address 0x1037

Register address 0x1031 (4151_{dez}) with a word count of 3	
Value	Configuration of Modbus Response Delay Time
Access	Read/write
Default	0x0000
Description	This register saves the value for the Modbus Response Delay Time for a Modbus connection. The time base is 1 ms. On the Modbus TCP connection, the response will be delayed by the inscribed time.

Table 149: Register address 0x1050

Register address 0x1050 (4176_{dec}) with a word count of 3 since Firmware version 9	
Value	Diagnosis of the connected I/O modules
Access	Read
Description	Diagnosis of the connected I/O modules, length 3 words Word 1: Number of the module Word 2: Number of the channel Word 3: Diagnosis

Table 150: Register address 0x2030

Register address 0x2030 (8240 _{dec}) with a word count of up to 65																
Value	Description of the connected I/O modules															
Access	Read module 0...64															
Description	Length 1...65 words These 65 registers identify the controller and the first 64 modules present in a node. Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 → Input module Bit position 1 → Output module Bit position 2...7 → Not used Bit position 8...14 → Module size in bits Bit position 15 → Designation digital module															
Examples:																
4 Channel Digital Input Module = 0x8401																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Code	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
Hex	8				4				0				1			
2 Channel Digital Output Module = 0x8202																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Code	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
Hex	8				2				0				2			

Table 151: Register address 0x2031

Register address 0x2031 (8241 _{dec}) with a word count of up to 64															
Value	Description of the connected I/O modules														
Access	Read modules 65...128														
Description	Length 1-64 words These 64 registers identify the 2nd block of I/O modules present (modules 65 to 128). Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 → Input module Bit position 1 → Output module Bit position 2...7 → Not used Bit position 8...14 → Module size in bits Bit position 15 → Designation digital module														

Table 152: Register address 0x2032

Register address 0x2032 (8242 _{dec}) with a word count of up to 64															
Value	Description of the connected I/O modules														
Access	Read modules 129...192														
Description	Length 1...64 words These 64 registers identify the 3rd block of I/O modules present (modules 129 to 192). Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 → Input module Bit position 1 → Output module Bit position 2...7 → Not used Bit position 8...14 → Module size in bits Bit position 15 → Designation digital module														

Table 153: Register address 0x2033

Register address 0x2033 (8243_{dec}) with a word count of up to 65	
Value	Description of the connected I/O modules
Access	Read modules 193 ... 255
Description	<p>Length 1-63 words These 63 registers identify the 4th block of I/O modules present (modules 193 to 255). Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below:</p> <p>Bit position 0 → Input module Bit position 1 → Output module Bit position 2...7 → Not used Bit position 8...14 → Module size in bits Bit position 15 → Designation digital module</p>

Table 154: Register address 0x2040

Register address 0x2040 (8256_{dec})	
Value	Implement a software reset
Access	Write (Write sequence 0xAA55 or 0x55AA)
Description	With Writing 0xAA55 or 0x55AA the register will be reset.

Table 155: Register address 0x2041

Register address 0x2041 (8257_{dez})	
Value	Flash Format
Access	Write (Write sequence 0xAA55 or 0x55AA)
Description	The file system Flash is again formatted.

Table 156: Register address 0x2042

Register address 0x2042 (8258_{dez})	
Value	Extract data files
Access	Write (Write sequence 0xAA55 or 0x55AA)
Description	The standard files (HTML pages) of the Coupler/Controller are extracted and written into the Flash.

Table 157: Register address 0x2043

Register address 0x2043 (8259_{dez})	
Value	0x55AA
Access	Write
Description	Factory Settings

12.2.5.5 Firmware Information Registers

The following registers contain information on the firmware of the controller:

Table 158: Register address 0x2010

Register address 0x2010 (8208 _{dec}) with a word count of 1	
Value	Revision, INFO_REVISION
Access	Read
Description	Firmware index, e.g. 0005 for version 5

Table 159: Register address 0x2011

Register address 0x2011 (8209 _{dec}) with a word count of 1	
Value	Series code, INFO_SERIES
Access	Read
Description	WAGO serial number, e.g. 0750 for WAGO-I/O-SYSTEM 750

Table 160: Register address 0x2012

Register address 0x2012 (8210 _{dec}) with a word count of 1	
Value	Order number, INFO_ITEM
Access	Read
Description	First part of WAGO order number, e.g. 841 for the controller 750-841 or 341 for the coupler 750-341 etc.

Table 161: Register address 0x2013

Register address 0x2013 (8211 _{dec}) with a word count of 1	
Value	Major sub item code, INFO_MAJOR
Access	Read
Description	Firmware version Major Revision

Table 162: Register address 0x2014

Register address 0x2014 (8212 _{dec}) with a word count of 1	
Value	Minor sub item code, INFO_MINOR
Access	Read
Description	Firmware version Minor Revision

Table 163: Register address 0x2020

Register address 0x2020 (8224 _{dec}) with a word count of up to 16	
Value	Description, INFO_DESCRIPTION
Access	Read
Description	Information on the controller, 16 words

Table 164: Register address 0x2021

Register address 0x2021 (8225_{dec}) with a word count of up to 8	
Value	Description, INFO_DESCRIPTION
Access	Read
Description	Time of the firmware version, 8 words

Table 165: Register address 0x2022

Register address 0x2022 (8226_{dec}) with a word count of up to 8	
Value	Description, INFO_DATE
Access	Read
Description	Date of the firmware version, 8 words

Table 166: Register address 0x2023

Register address 0x2023 (8227_{dec}) with a word count of up to 32	
Value	Description, INFO_LOADER_INFO
Access	Read
Description	Information to the programming of the firmware, 32 words

12.2.5.6 Constant Registers

The following registers contain constants, which can be used to test communication with the master:

Table 167: Register address 0x2000

Register address 0x2000 (8192 _{dec})	
Value	Zero, GP_ZERO
Access	Read
Description	Constant with zeros

Table 168: Register address 0x2001

Register address 0x2001 (8193 _{dec})	
Value	Ones, GP_ONES
Access	Read
Description	Constant with ones <ul style="list-style-type: none"> • -1 if this is declared as "signed int" • MAXVALUE if it is declared as "unsigned int"

Table 169: Register address 0x2002

Register address 0x2002 (8194 _{dec})	
Value	1,2,3,4, GP_1234
Access	Read
Description	This constant value is used to test the Intel/Motorola format specifier. If the master reads a value of 0x1234, then with Intel format is selected – this is the correct format. If 0x3412 appears, Motorola format is selected.

Table 170: Register address 0x2003

Register address 0x2003 (8195 _{dec})	
Value	Mask 1, GP_AAAA
Access	Read
Description	This constant is used to verify that all bits are accessible to the fieldbus master. This will be used together with register 0x2004.

Table 171: Register address 0x2004

Register address 0x2004 (8196 _{dec})	
Value	Mask 1, GP_5555
Access	Read
Description	This constant is used to verify that all bits are accessible to the fieldbus master. This will be used together with register 0x2003.

Table 172: Register address 0x2005

Register address 0x2005 (8197 _{dec})	
Value	Maximum positive number, GP_MAX_POS
Access	Read
Description	Constant in order to control arithmetic.

Table 173: Register address 0x2006

Register address 0x2006 (8198_{dec})	
Value	Maximum negative number, GP_MAX_NEG
Access	Read
Description	Constant in order to control arithmetic

Table 174: Register address 0x2007

Register address 0x2007 (8199_{dec})	
Value	Maximum half positive number, GP_HALF_POS
Access	Read
Description	Constant in order to control arithmetic

Table 175: Register address 0x2008

Register address 0x2008 (8200_{dec})	
Value	Maximum half negative number, GP_HALF_NEG
Access	Read
Description	Constant in order to control arithmetic

Table 176: Register address 0x3000 to 0x5FFF

Register address 0x3000 to 0x5FFF (12288_{dec} to 24575_{dec})	
Value	Retain range
Access	Read/write
Description	These registers can be accessed as the flag/retain range

12.3 EtherNet/IP (Ethernet/Industrial Protocol)

12.3.1 General

EtherNet/IP stands for Ethernet Industrial Protocol and defines an open industry standard that extends the classic Ethernet with an industrial protocol. This standard was jointly developed by ControlNet International (CI) and the Open DeviceNet Vendor Association (ODVA) with the help of the Industrial Ethernet Association (IEA).

This communication system enables devices to exchange time-critical application data in an industrial environment. The spectrum of devices ranges from simple I/O devices (e.g., sensors) through to complex controllers (e.g., robots).

EtherNet/IP is based on the TCP/IP protocol family and consequently uses the bottom 4 layers of the OSI layer model in unaltered form so that all standard Ethernet communication modules such as PC interface cards, cables, connectors, hubs and switches can also be used with EtherNet/IP.

Positioned above the transport layer is the encapsulation protocol, which enables use of the Control & Information Protocol (CIP) on TCP/IP and UDP/IP.

CIP, as a major network independent standard, is already used with ControlNet and DeviceNet. Therefore, converting from one of these protocols to EtherNet/IP is easy to do. Data exchange takes place with the help of an object model.

In this way, ControlNet, DeviceNet and EtherNet/IP have the same application protocol and can therefore jointly use device profiles and object libraries. These objects enable plug-and-play interoperability between complex devices of different manufacturers.

12.3.2 Protocol overview in the OSI model

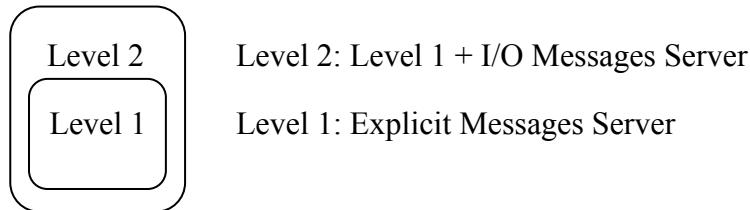
In order to clarify the interrelationships between DeviceNet, ControlNet and EtherNet/IP, the following diagram presents the associated ISO/OSI reference model.

Table 177: ISO/OSI reference model

7 Application Layer	Object Library (Communications, Applications, Time Synchronization)		Safety Object Library	Comon Industrial Protocol (CIP)
6 Presentation Layer	Data Management Services Explicit and I/O Messages		Safety Services and Messages	
5 Session Layer	Connection Management, Routing			
4 Transport Layer	TCP/UDP	CompoNet Network and Transport	ControlNet Network and Transport	Network Adaptations of CIP
3 Network Layer	Internet Protocol			
2 Data Link Layer	Ethernet CSMA/CD	CompoNet Time Slot	ControlNet CTDMA	CAN CSMA/NBA
1 Physical Layer	Ethernet	CompoNet	ControlNet	DeviceNet

12.3.3 Characteristics of the EtherNet/IP Protocol Software

The Ethernet/IP product classes are divided into 4 levels with each level containing a particular functionality. Each higher level in turn possesses at least the functionality of a lower level. The fieldbus coupler supports levels 1 and 2 of the Ethernet/IP product classes, which immediately build on each other.



- Unconnected Message Manager (UCMM) client and server
- 128 Encapsulation Protocol sessions
- 128 Class 3 or Class 1 connections combined
 - Class 3 connection – explicit messages (connection oriented, client and server)
 - Class 1 connection – I/O messages (connection oriented, client and server)

12.3.4 EDS File

The "Electronic Data Sheets" file (EDS file for short) contains the characteristics of the fieldbus coupler/controller and information regarding its communication capabilities. The EDS file required for Ethernet/IP operation is imported and installed by the corresponding configuration software.

Note



Downloading the EDS file!

You can download the EDS file in the download area of the WAGO web site:
<http://www.wago.com> → Service → Downloads → AUTOMATION

Information



Information about installing the EDS file

When installing the EDS file, refer to the information provided in the documentation of the configuration software, which you are using.

12.3.5 Object Model

12.3.5.1 General

For network communication, Ethernet/IP utilizes an object model in which all functions and data of a device are described.

Each node in the network is depicted as a collection of objects.

The object model contains terms that are defined as follows:

Object:

An object is an abstract representation of individual, related components within a device. It is determined by its data or attributes, its outwardly applied functions or services, and by its defined behavior.

Class:

A class describes a series of objects which all represent the same type of system components. A class is the generalization of an object. All objects in a class are identical as regards form and behavior, but can comprise differing attribute values.

Instance:

An instance describes a specific and physical occurrence of an object. The terms "object," "instance" and "object instance" all refer to a specific instance. Different instances of a class have the same services, the same behavior and the same variables (attributes). However, you can have different variable values. For example, Finland is an instance of the "Land" object class.

Variable:

The variables (attributes) describe an externally visible characteristic or the function of an object. Typical attributes include configuration or status information.

For example, the ASCII name of an object or the repetition frequency of a periodic object is output.

Service:

A service is a function supported by an object and/or an object class. CIP defines a group of common services that are applied to the attributes. These services execute specified actions.

Example: Reading variables.

Behavior:

The behavior specifies how an object functions. The functions result from various occurrences, which are determined by the object, e.g. receiving service requests, recording internal errors or the sequence of timers.

12.3.5.2 Class Overview

CIP classes are included in the CIP specification of ODVA. They describe the properties (Volume 1, "Common Industrial Protocol") of Ethernet and CAN independent of their physical interface. The physical interface is described in a separate specification. For Ethernet/IP, this is Volume 2 ("Ethernet/IP Adaptation of CIP"), which describes the adaption of Ethernet /IP to CIP.

For this purpose, WAGO uses classes 01_{hex}, 02_{hex}, 04_{hex}, 05_{hex}, 06_{hex} and F4_{hex}, which are described in Volume 1 ("Common Industrial Protocol"). Classes F5_{hex} and F6_{hex} are supported from Volume 2 ("Ethernet/IP Adaptation of CIP").

WAGO-specific classes listed in the overview table below are also available.

All CIP Common classes listed and the WAGO-specific classes listed below that are described in detail in the following individual sections after a brief explanation of the table headings in the object descriptions.

Table 178: CIP common class



Class	Name
01 _{hex}	Identity
02 _{hex}	Message Router
04 _{hex}	Assembly
05 _{hex}	Connection
06 _{hex}	Connection Manager
F5 _{hex}	TCP/IP Interface Object
F6 _{hex}	Ethernet Link Object

Table 179: WAGO specific classes

Class	Name
64 _{hex}	Coupler/Controller Configuration Object
65 _{hex}	Discrete Input Point
66 _{hex}	Discrete Output Point
67 _{hex}	Analog Input Point
68 _{hex}	Analog Output Point
69 _{hex}	Discrete Input Point Extended 1
6A _{hex}	Discrete Output Point Extended 1
6B _{hex}	Analog Input Point Extended 1
6C _{hex}	Analog Output Point Extended 1
6D _{hex}	Discrete Input Point Extended 2
6E _{hex}	Discrete Output Point Extended 2
6F _{hex}	Analog Input Point Extended 2
70 _{hex}	Analog Output Point Extended 2
71 _{hex}	Discrete Input Point Extended 3
72 _{hex}	Discrete Output Point Extended 3
73 _{hex}	Analog Input Point Extended 3
74 _{hex}	Analog Output Point Extended 3
80 _{hex}	Module Configuration
81 _{hex}	Module Configuration Extended 1
A0 _{hex}	Input fieldbus variable USINT
A1 _{hex}	Input fieldbus variable USINT Extended 1
A2 _{hex}	Input fieldbus variable USINT Extended 2
A3 _{hex}	Output fieldbus variable USINT
A4 _{hex}	Output fieldbus variable USINT Extended 1
A5 _{hex}	Output fieldbus variable USINT Extended 2
A6 _{hex}	Input fieldbus variable UINT
A7 _{hex}	Input fieldbus variable UINT Extended 1
A8 _{hex}	Output fieldbus variable UINT
A9 _{hex}	Output fieldbus variable UINT Extended 1
AA _{hex}	Input fieldbus variable UDINT
AB _{hex}	Input fieldbus variable UDINT Offset UINT
AC _{hex}	Output fieldbus variable UDINT
AD _{hex}	Output fieldbus variable UDINT Offset UINT

12.3.5.3 Explanation of the Table Headings in the Object Descriptions

Table 180: Explanation of the table headings in the object descriptions

Table heading	Description
Attribute ID	Integer value which is assigned to the corresponded attribute
Access	<p>Set: The attribute can be accessed by means of Set_Attribute services.</p> <p> Note Response also possible with Get_Attribute service! All the set attributes can also be accessed by means of Get_Attribute services.</p> <p>Get: The attribute can be accessed by means of Get_Attribute services.</p> <p>Get_Attribute_All: Delivers content of all attributes.</p> <p>Set_Attribute_Single: Modifies an attribute value.</p> <p>Reset: Performs a restart. 0: Restart 1: Restart and restoration of factory settings</p>
NV	<p>NV (non volatile): The attribute is permanently stored in the controller.</p>
	<p>V (volatile): The attribute is not permanently stored in the controller.</p> <p> Note Without specifying, the attribute is not saved! If this column is missing, all attributes have the type V (volatile).</p>
Name	Designation of the attribute
Data type	Designation of the CIP data type of the attribute
Description	Short description for the Attribute
Default value	Factory settings

12.3.5.4 Identity (01_{hex})

The "Identity" class provides general information about the fieldbus coupler/controller that clearly identifies it.

Instance 0 (Class Attributes)

Table 181: Identity (01_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum instance	1 (0x0001)
3	Get	Max ID number of class attributes	UINT	Maximum number of class attributes	0 (0x0000)
4	Get	Max ID number of instance attribute	UINT	Maximum number of instance attributes	0 (0x0000)

Instance 1

Table 182: Identity (01_{hex}) – Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Vendor ID	UINT	Manufacturer identification	40 (0x0028)
2	Get	Device Type	UINT	General type designation of the product	12 (0x000C)
3	Get	Product Code	UINT	Designation of the coupler/ controller	z. B. 841 (0x0349), 873 (0x0369), 341(0x0155) etc.
4	Get	Revision	STRUCT of:	Revision of the identity objects	Depending on the firmware
		Major Revision	UINT		
		Minor Revision	UINT		
5	Get	Status	WORD	Current status of the device	Bit 0 Assignment to a master Bit 1 = 0 reserved Bit 2 (configured) = 0 Configuration is unchanged = 1 Configuration is different to the manufacturers parameters Bit 3 = 0 reserved Bit 4-7 Extended Device Status =0010 at least one faulted I/O connection =0011 no I/O connection established Bit 8-11 not used Bit 12-15 =0 reserved
6	Get	Serial Number	UINT	Serial number	The last 4 digits of MAC ID
7	Get	Product Name	SHORT STRING	Product name	

Common Services

Table 183: Identity (01_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
01 _{hex}	Yes	Yes	Get_Attribute_All	Supplies contents of all attributes
05 _{hex}	No	Yes	Reset	Implements the reset service
				Service parameter 0: Emulates a Power On reset 1: Emulates a Power On reset and re-establishes factory settings
0E _{hex}	No	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.5 Message Router (02_{hex})

The "Message Router Object" provides connection points (in the form of classes or instances), which can use a client for addressing services (reading, writing). These messages can be transmitted both when connected and when unconnected from the client to the fieldbus coupler.

Instance 0 (Class Attributes)

Table 184: Message router (02_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Number of Attributes	UINT	Number of attributes	0 (0x0000)
3	Get	Number of Services	UINT	Number of services	0 (0x0000)
4	Get	Max ID Number of Class Attributes	UINT	Maximum number of class attributes	0 (0x0000)
5	Get	Max ID Number of Instance Attributes	UINT	Maximum number of instance attributes	0 (0x0000)



Note

Get_Attribute_All service can only be used!

The class attributes are only accessible with the Get_Attribute_All service.

Instance 1

Table 185: Message router (02_{hex}) – Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	ObjectList	STRUCT of:	-	
		Number	UINT	Number of implemented classes	40 (0x0028)
		Classes	UINT	Implemented classes	01 00 02 00 04 00 06 00 F4 00 F5 00 F6 00 64 00 65 0066 0067 00 68 00 69 00 6A 00 6B 00 6C 00 6D 00 6E 00 6F 00 70 00 71 00 72 00 73 00 74 00 80 00 81 00 A0 00 A1 00 A2 00 A6 00 A7 00 AA 00 AB 00 A3 00 A4 00 A5 00 A8 00 A9 00 AC 00 AD 00
2	Get	NumberAvailable	UINT	Maximum number of different connections	128 (0x0080)

Common Services

Table 186: Message router (02_{hex}) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
01 _{hex}	Yes	No	Get_Attribute_All	Supplies contents of all attributes
0E _{hex}	No	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.6 Assembly Object (04_{hex})

Using the "Assembly" classe, even several diverse objects can be combined. These could be, for example, input and output data, status and control information or diagnostic information. WAGO uses the manufacturer-specific instances in order to provide these objects for you in various arrangements. This gives you an efficient way to exchange process data. The following is a description of the individual static Assembly instances with their contents and arrangements.

Instance (Class Attributes)

Table 187: Assembly (04_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	2 (0x0002)
2	Get	Max Instance	UINT	Highest Instance	111 (0x006F)

Overview of static Assembly instances

Table 188: Overview of static Assembly instances

Instance	Description
Instance 101 (65 _{hex})	For analog and digital output data, as well as fieldbus input variables
Instance 102 (66 _{hex})	For digital output data and fieldbus input variables
Instance 103 (67 _{hex})	For analog output data and fieldbus input variables
Instance 104 (68 _{hex})	For analog and digital input data, status and fieldbus output variables
Instance 105 (69 _{hex})	For digital input data, status and fieldbus output variables
Instance 106 (6A _{hex})	For analog input data, status and fieldbus output variables
Instance 107 (6B _{hex})	For digital and analog input data and fieldbus output variables
Instance 108 (6C _{hex})	For digital input data and fieldbus output variables
Instance 109 (6D _{hex})	For analog input data and fieldbus output variables
Instance 110 (6E _{hex})	For fieldbus output variables
Instance 111 (6F _{hex})	For fieldbus input variables

Instance 101 (65_{hex})

This assembly instance contains analog and digital output data. Any fieldbus input variables that may be defined are attached behind this.

Table 189: Static assembly instances – Instance 101 (65_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get/Set	Data	ARRAY of BYTE	Only analog and digital output data, as well as possible fieldbus input variables, are contained in the process image.	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 102 (66_{hex})

This assembly instance contains digital output data and fieldbus input variables only.

Table 190: Static assembly instances – Instance 102 (66_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get/Set	Data	ARRAY of BYTE	Digital output data and fieldbus input variables are contained in the process image.	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 103 (67_{hex})

This assembly instance contains analog output data and fieldbus input variables only.

Table 191: Static assembly instances – Instance 103 (67_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get/Set	Data	ARRAY of BYTE	Analog output data and fieldbus input variables are contained in the process image.	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 104 (68_{hex})

This assembly instance contains analog and digital input data, status (= value from class 100, instance 1, attribute 5) and fieldbus output variables.

Table 192: Static assembly instances – Instance 104 (68_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Analog and digital input data, the status and fieldbus output variables are contained in the process image.	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 105 (69_{hex})

This assembly instance contains only digital input data, status (= value from class 100, instance 1, attribute 5) and fieldbus output variables.

Table 193: Static assembly instances – Instance 105 (69_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Digital input data, status and fieldbus output variables are contained in the process image	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 106 (6A_{hex})

This assembly instance contains only analog input data, status (= value from class 100, instance 1, attribute 5) and fieldbus output variables.

Table 194: Static assembly instances – Instance 106 (6A_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Analog input data, status and fieldbus output variables are contained in the process image.	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 107 (6B_{hex})

This assembly instance contains analog and digital input data and fieldbus output variables.

Table 195: Static assembly instances – Instance 107 (6B_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Analog and digital input data and fieldbus output variables are contained in the process image.	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 108 (6C_{hex})

This assembly instance contains only digital input data and fieldbus output variables.

Table 196: Static assembly instances – Instance 108 (6C_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Digital input data and fieldbus output variables are contained in the process image.	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 109 (6D_{hex})

This assembly instance contains only analog input data and fieldbus output variables.

Table 197: Static assembly instances – Instance 109 (6D_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Analog input data and fieldbus output variables are contained in the process image.	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 110 (6E_{hex})

This assembly instance contains fieldbus output variables.

Table 198: Static assembly instances – Instance 110 (6E_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Reference of the process image: only PFC output variables	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 111 (6F_{hex})

This assembly instance contains fieldbus input variables only.

Table 199: Static assembly instances – Instance 111 (6F_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Set	Data	ARRAY of BYTE	Reference of the process image: only PFC input variables	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 198 (C6_{hex}) “Input Only”

This instance is used to establish a connection when no outputs are to be addressed or when inputs, which are already being used in an exclusive owner connection, are to be interrogated. The data length of this instance is always zero. This instance can only be used in the “consumed path” (seen from the slave device).

Instance 199 (C7_{hex}) “Listen only”

This instance is used to establish a connection based on an existing exclusive owner connection. The new connection also has the same transmission parameters as the exclusive owner connection. When the exclusive owner connection is cleared, this connection, too, is automatically cleared. The data length of this instance is always zero.

This instance can only be used in the “consumed path” (from the point of view of the slave device).

Common Service

Table 200: Static assembly instances – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

The software inspects the writing of attribute 3 of assembly instances 101, 102 and 103. If the limit value has been exceeded, it is identified and, if necessary, corrected. However, a write request is not rejected. This means that if less data is received than expected, only this data is written. If more data is received than expected, the received data at the upper limit is deleted. In the case of explicit messages, however, a defined CIP is generated even though the data has been written.

12.3.5.7 Connection (05_{hex})

Because the connections are established and terminated via the connection manager, the class and instance attributes of this class are not visible.

12.3.5.8 Connection Manager (06_{hex})

The "Connection Manager Object" provides the internal resources that are required for the input and output data and explicit messages. In addition, the administration of this resource is an assignment of the "Connection Manager Object".

For each connection (input and output data or explicit), another instance of the connection class is created. The connection parameters are extracted from the "Forward Open" service, which is responsible for establishing a connection.

The following services are supported for the first instance:

- Forward_Open
- Unconnected_Send
- Forward_Close

No class and instance attributes are visible.

12.3.5.9 Port Class (F4_{hex})

The "Port Class Object" specifies the existing CIP ports on the fieldbus coupler/coupler. There is one instance for each CIP port.

Instance 0 (Class Attributes)

Table 201: Port class (F4_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	1 (0x0001)
3	Get	Num Instances	UINT	Number of current ports	1 (0x0001)
8	Get	Entry Port	UINT	Instance of the port object where the request arrived.	1 (0x0001)
9	Get	All Ports	Array of Struct UINT	Array with instance attributes 1 and 2 of all instances	0 (0x0000) 0 (0x0000) 4 (0x0004) 2 (0x0002)

Instance 1

Table 202: Port class (F4_{hex}) – Instance 1

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Port Type	UINT	-	4 (0x0004)
2	Get	V	Port Number	UINT	CIP port number	2 (0x0002) (EtherNet/IP)
3	Get	V	Port Object	UINT	Number of 16 bit words in the following path	2 (0x0002)
				Padded EPATH	Object, which manages this port	0x20 0xF5 0x24 0x01 (equals TCP/IP Interface Object)
4	Get	V	Port Name	SHORT_STRING	Port name	""
7	Get	V	Node Address	Padded EPATH	Port segment (IP address)	Depends on IP address

Common Services

Table 203: Port class (F4_{hex}) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
01 _{hex}	Yes	Yes	Get_Attribute_All	Supplies contents of all attributes
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.10 TCP/IP Interface (F5_{hex})

The "TCP/IP Interface Object" provides for the configuration of the TCP/IP network interface of a fieldbus coupler/controller. Examples of configurable objects include the IP address, the network mask and the gateway address of the fieldbus coupler/controller.

The underlying physical communications interface that is connected with the TCP/IP interface object can be any interface supported by the TCP/IP protocol. Examples of components that can be connected to a TCP/IP interface object include the following: an Ethernet interface 802.3, an ATM (Asynchronous Transfer Mode) interface or a serial interface for protocols such as PPP (Point-to-Point Protocol).

The TCP/IP interface object provides an attribute, which is identified by the link-specific object for the connected physical communications interface. The link-specific object should typically provide link-specific counters as well as any link-specific configuration attributes.

Each device must support exactly one instance of the TCP/IP interface object for each TCP/IP-compatible communications interface. A request for access to the first instance of the TCP/IP interface object must always refer to the instance connected with the interface, which is used to submit the request.

Instance 0 (Class Attributes)

Table 204: TCP/IP interface (F5_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	1 (0x0001)
3	Get	Num Instances	UINT	Number of the current instanced connections	1 (0x0001)

Instance 1

Table 205: TCP/IP interface (F5_{hex}) – Instance 1

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Status	DWORD	Interface state	-
2	Get	V	Configuration Capability	DWORD	Interface flags for possible kinds of configuration	0x00000017
3	Set	NV	Configuration Control	DWORD	Specifies, how the device gets is TCP/IP configuration after the first Power On	0x00000011
4	Get	V	Physical Link Object	STRUCT of		
			Path size	UINT	Number of 16 Bit words in the following path	0x0002
			Path	Padded EPATH	Logical path, which points to the physical Link object	0x20 0xF6 0x24 0x03 (equates to the Ethernet Link Object)
5	Set	NV	Interface Configuration	STRUCT of	-	
			IP Address	UDINT	IP address	0
			Network Mask	UDINT	Net work mask	0
			Gateway Address	UDINT	IP address of default gateway	0
			Name Server	UDINT	IP address of the primary name of the server	0
			Name Server 2	UDINT	IP address of the secondary name of the server	0
			Domain Name	STRING	Default domain name	""
6	Set	NV	Host Name	STRING	Device name	""

Common Services

Table 206: TCP/IP interface (F5_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
01 _{hex}	Yes	Yes	Get_Attribute_All	Supplies contents of all attributes
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.11 Ethernet Link (F6_{hex})

The "Ethernet Link Object" contains link-specific counter and status information for an Ethernet 802.3 communications interface. Each device must support exactly one instance of the Ethernet Link Object for each Ethernet IEEE 802.3 communications interface on the module. An Ethernet link object instance for an internal interface can also be used for the devices, e.g. an internal port with an integrated switch.

Instance 0 (Class Attributes)Table 207: Ethernet link (F5_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	3 (0x0003)
2	Get	Max Instance	UDINT	Max. number of instances	3 (0x0003)
3	Get	Num Instances	UDINT	Number of the current instanced connections	3 (0x0003)

Instance 1

Table 208: Ethernet link (F6_{hex}) – Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Interface Speed	UDINT	Transfer rate	10 (0x0A) or 100 (0x64)
2	Get	Interface Flags	DWORD	Interface configuration and status information Bit 0: Link status Bit 1: Half/full duplex Bit 2...4: Detection status Bit 5: Manual settings require reset Bit 6: Local hardware error Bit 7...31: Reserved	Value is dependent upon Ethernet connection.
3	Get	Physical Address	ARRAY of 6 UINTs	MAC layer address	MAC ID of the device
6	Set	Interface Control	STRUCT of:	Configuration of the physical interface	-
		Control Bits	WORD	Interface configuration bits Bit 0: Automatic detection Bit 1: Default duplex mode Bit 2...15: Reserved	0x0001
		Forced Interface Speed	UINT	Preset interface speed	10 (0x000A) or 100 (0x0064)
7	Get	Interface Type	USINT	Interface type Value 0: Unknown Value 1: Internal interface; e.g., in the case of an integrated switch Value 2: Twisted pair (e.g. 100Base-TX). Value 3: fiber glass (e.g. 100Base-FX). Value 4...256: Reserved	2 (0x02) – Twisted Pair
8	Get	Interface Status	USINT	Interface status Value 0: Unknown Value 1: Interface active and ready to send/receive. Value 2: Interface deactivated. Value 3: Interface is testing Wert 4...256: Reserved	-

Table 208: Ethernet link (F6_{hex}) – Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
9	Get/ Set	Admin Status	USINT	Admin status: Value 0: Reserved Value 1: Interface active Value 2: Interface deactivated. Is this the only CIP interface, a request for deactivation will be receipted with error code 0x09 Value 3...256: Reserved	1 (0x01)
10	Get	Interface Label	SHORT_STRING	Name of the interface	“Port 1“

Instance 2 – Port 2

Table 209: Ethernet link (F6_{hex}) – Instance 2

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Interface Speed	UDINT	Transfer rate	10 (0x0000000A) or 100 (0x00000064)
2	Get	Interface Flags	DWORD	Interface configuration and status information Bit 0: Link status Bit 1: Half/full duplex Bit 2...4: Detection status Bit 5: Manual settings require reset Bit 6: Local hardware error Bit 7...31: Reserved	Value is dependent upon Ethernet connection.
3	Get	Physical Address	ARRAY of 6 UINTs	MAC layer address	MAC-ID des Fieldbus couplers/ controllers
6	Set	Interface Control	STRUCT of:	Configuration of the physical interface	-
		Control Bits	WORD	Interface configuration bits Bit 0: Automatic detection Bit 1: Default duplex mode Bit 2...15: Reserved	0x0001
		Forced Interface Speed	UINT	Preset interface speed	10 (0x000A) or 100 (0x0064)
7	Get	Interface Type	USINT	Interface type Value 0: Unknown Value 1: Internal interface; e.g., in the case of an integrated switch Value 2: Twisted pair (e.g. 100Base-TX). Value 3: fiber glass (e.g. 100Base-FX). Value 4...256: Reserved	2 (0x02) – Twisted Pair
8	Get	Interface Status	USINT	Interface status Value 0: Unknown Value 1: Interface active and ready to send/receive. Value 2: Interface deactivated. Value 3: Interface is testing Wert 4...256: Reserved	-

Table 209: Ethernet link (F6_{hex}) – Instance 2

Attribute ID	Access	Name	Data type	Description	Default value
9	Get/ Set	Admin Status	USINT	Admin status: Value 0: Reserved Value 1: Interface active Value 2: Interface deactivated. Is this the only CIP interface, a request for deactivation will be receipted with error code 0x09 Value 3...256: Reserved	1 (0x01)
10	Get	Interface Label	SHORT_STRING	Name of the interface	“Port 2“

Instance 3 – Internal Port 3

Table 210: Ethernet link (F6_{hex}) – Instance 3

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Interface Speed	UDINT	Transfer rate	100 (0x64)
2	Get	Interface Flags	DWORD	Interface configuration and status information	3 (0x03) – Link active (Bit 0), Full duplex (Bit 1)
3	Get	Physical Address	ARRAY of 6 UINTs	MAC layer address	MAC ID of the device
6	Set	Interface Control	STRUCT of:	Configuration of the physical interface	-
		Control Bits	WORD	Interface configuration bits	3 (0x03) – Link active (Bit 0), Full duplex (Bit 1)
		Forced Interface Speed	UINT	Baud rate	100 (0x64)
7	Get	Interface Type	UINT	Interface type	1 (0x01) – internal Port
8	Get	Interface Status	UINT	Interface status	1 (0x01) – active
9	Get	Admin Status	UINT	Admin status	1 (0x01) – active
10	Get	Interface Label	SHORT_STRING	Name of the interface	„Internal Port 3“

Common Services

Table 211: Ethernet link (F6_{hex}) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
01 _{hex}	Yes	Yes	Get_Attribute_All	Supplies contents of all attributes
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value



Note

Changes with service "Set_Attribute_Single" not directly effective!
Attributes (particularly the attributes 6 and 9) which were changed over the service "Set_Attribute_Single", become only effective after the next Power-On-Reset of the controller.

12.3.5.12 Coupler/Controller Configuration (64_{hex})

The fieldbus coupler configuration class allows reading and configuration of some important fieldbus/controller process parameters. The following listings explain in details all supported instances and attributes.

Instance 0 (Class Attributes)Table 212: Coupler/Controller configuration (64_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	1 (0x0001)

Instance 1Table 213: Coupler/Controller configuration (64_{hex}) – Instance 1

Attribute ID	Access	NV	Name	Data type	Description	Default value
5 (0x05)	Get	V	ProcessState	USINT	State of coupler/controller, error mask: Bit 0: Internal bus error Bit 3: Module diagnostics (0x08) Bit 7: Fieldbus error (0x80)	0
6 (0x06)	Get	V	DNS_i_TrmnIdia	UINT	Module diagnostics: Bit 0..7: Module number Bit 8..14: Module channel Bit 15: 0/1 Error, repair/arisen	0
7 (0x07)	Get	V	CnfLen.AnalogOut	UINT	Number of I/O bits for the analog output	-
8 (0x08)	Get	V	CnfLen.AnalogInp	UINT	Number of I/O bits for the analog input	-
9 (0x09)	Get	V	CnfLen.DigitalOut	UINT	Number of I/O bits for the digital output	-
10 (0x0A)	Get	V	CnfLen.DigitalInp	UINT	Number of I/O bits for the digital input	-
11 (0x0B)	Set	NV	Bk_Fault_Reaction	USINT	Fieldbus error reaction 0: stop local I/O cycles 1: set all output to 0 2: no error reaction 3: no error reaction 4: PFC task takes over control of the outputs (apply to controllers)	1
12..26 (0x0C...0x1A)	Reserved for compatibility to DeviceNet					
40..43 (0x28...0x2B)	Reserved for compatibility to DeviceNet					
45 (0x2D)	Get	V	Bk_Led_Err Code	UINT	I/O LED error code	0
46 (0x2E)	Get	V	Bk_Led_Err Arg	UINT	I/O LED error argument	0

Attribute ID	Access	NV	Name	Data type	Description	Default value
47 (0x2F)	Get	V	Bk_Diag_Value	UINT	Contains the diagnostic byte Note: This attribute has to be read out before attribute 6 (DNS_i_Trmnldia), because during the reading of attribute 6 the diagnostic byte contains the data of the next diagnostic	0
100 (0x64)	Set	NV	Bk_FbInp_Var_Cnt	UINT	Determines the number of bytes for the PFC input fieldbus variables, which are added to the assembly object. This number is added to the consuming path. assembly instances (101...103)	0
101 (0x65)	Set	NV	Bk_FbOut_Var_Cnt	UINT	Determines the number of bytes for the PFC output fieldbus variables, which are added to the assembly object. This number is added to the producing path. assembly instances (104...109)	0
102 (0x66)	Set	NV	Bk_FbInp_Plc_Only_Var_Cnt	UINT	Determines the number of bytes for the PFC input fieldbus variables, which are received via assembly instance 111.	4
103 (0x67)	Set	NV	Bk_FbInp_Start_Plc_Var_Cnt	UINT	Determines starting from which position the PFC input fieldbus variables for the assembly instance 111 to be received.	0
104 (0x68)	Set	NV	Bk_FbOut_Plc_Only_Var_Cnt	UINT	Determines the number of bytes for the PFC output fieldbus variables, which are received via assembly instance 110.	4
105 (0x69)	Set	NV	Bk_FbOut_Start_Plc_Var_Cnt	UINT	Determines starting from which position the PFC output fieldbus variables for the assembly instance 110 to be received.	0

120 (0x78)	Set	NV	Bk_Header CfgOT	UINT	Indicates whether the RUN/IDLE header is used originator → target direction 0: is used 1: is not used	0x0000
121(0x79)	Set	NV	Bk_Header CfgTO	UINT	Indicates whether the RUN/IDLE header is used originator → target direction 0: is used 1: is not used	0x0001

Common Service

Table 214: Coupler/Controller configuration (64_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.13 Discrete Input Point (65_{hex})

This class allows the reading of data of a particular digital input point.

Instance 0 (Class-Attributes)

Table 215: Discrete input point (65_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 1 ... 255 (Digital output value 1 up to 255)

Table 216: Discrete input point (65_{hex}) – Instance 1...255

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	DipObj_Value	BYTE	Digital output (only Bit 0 is valid)	-

Common Services

Table 217: Discrete input point (65_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.14 Discrete Input Point Extended 1 (69_{hex})

The extension of the "Discrete Input Point" class enables the reading of data from a fieldbus node that contains over 255 digital input points (DIPs). The instance

scope of the "Discrete Input Point Extended 1" class covers DIPs from 256 to 510 in the fieldbus node.

Instance 0 (Class Attributes)

Table 218: Discrete Input Point Extended 1(69 hex.) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 256 ... 510 (Digital input value 256 up to 510)

Table 219: Discrete output point (66 hex) – Instance 256...510

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	DipObj_Value	BYTE	Digital input (only Bit 0 is valid)	-

Common Services

Table 220: Discrete Input Point Extended 1 (69 hex) – Common service

Service code	Service available		Service-name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.15 Discrete Input Point Extended 2 (6D hex)

The extension of the "Discrete Input Point" class enables the reading of data from a fieldbus node that contains over 510 digital input points (DIPs). The instance scope of the "Discrete Input Point Extended 2" class covers DIPs from 511 to 765 in the fieldbus node.

Instance 0 (Class Attributes)

Table 221: Discrete Input Point Extended 2 (6D hex) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 511 ... 765 (Digital input value 511 up to 765)

Table 222: Analog input point (67 hex) – Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	ARRAY of BYTE	Analog input	-
2	Get	AipObj_Value_Length	USINT	Length of the input data AipObj_Value (in byte)	-

Common Services

Table 223: Analog input point (67_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.16 Discrete Input Point Extended 3 (71_{hex})

The extension of the "Discrete Input Point" class enables the reading of data from a fieldbus node that contains over 765 digital input points (DIPs). The instance scope of the "Discrete Input Point Extended 3" class covers DIPs from 766 to 1020 in the fieldbus node.

Instance 0 (Class-Attributes)

Table 224: Discrete Input Point Extended 3 (71_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 766 ... 1020 (Digital input value 766 up to 1020)

Table 225: Discrete Input Point Extended 3 (71_{hex}) – Instance 766...1020

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	DipObj_Value	BYTE	Digital input (only Bit 0 is valid)	-

Common Services

Table 226: Discrete Input Point Extended 3 (71_{hex}) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.17 Discrete Output Point (66_{hex})

This class enables data exchange for a particular digital output point.

Instance 0 (Class Attributes)

Table 227: Discrete Output Point (66_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 1 ... 255 (Digital output value 1 up to 255)

Table 228: Discrete Output Point (66_{hex}) – Instance 1...255

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	DopObj_Value	BYTE	Digital Output (only Bit 0 valid)	-

Common Services

Table 229: Discrete Output Point (66_{hex}) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.18 Discrete Output Point Extended 1 (6A_{hex})

The extension of the "Discrete Output Point" class enables the exchange of data from a fieldbus node that contains over 255 digital output points (DOPs). The instance scope of the "Discrete Output Point Extended 1" class covers DOPs from 256 to 510 in the fieldbus node.

Instance 0 (Class Attributes)

Table 230: Discrete Output Point Extended 1 (6A_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 256 ... 510 (Digital output value 256 up to 510)

Table 231: Discrete Output Point Extended 1 (6A_{hex}) – Instance 256...510

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	DopObj_Value	BYTE	Digital Output (only Bit 0 valid)	-

Common Services

Table 232: Discrete Output Point Extended 1 (6A_{hex}) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.19 Discrete Output Point Extended 2 (6E_{hex})

The extension of the "Discrete Output Point" class enables the exchange of data from a fieldbus node that contains over 510 digital output points (DOPs). This instance cope of the "Discrete Output Point Extended 1" class covers the DOPs from 511 to 765 in the fieldbus node.

Instance 0 (Class Attributes)Table 233: Discrete Output Point Extended 2 (6E_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 511 ... 765 (Digital output value 511 up to 765)Table 234: Discrete Output Point Extended 2 (6E_{hex}) – Instance 511...765

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	DopObj_Value	BYTE	Digital Output (only Bit 0 valid)	-

Common ServicesTable 235: Discrete Output Point Extended 2 (6E_{hex}) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.20 Discrete Output Point Extended 3 (72_{hex})

The extension of the "Discrete Output Point" class enables the exchange of data from a fieldbus node that contains over 765 digital output points (DOPs). The instance scope of the "Discrete Output Point Extended 2" class covers DOPs from 766 to 1020 in the fieldbus node.

Instance 0 (Class Attributes)Table 236: Discrete Output Point Extended 3 (72_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 766 ... 1020 (Digital Output value 766 up to 1020)

Table 237: Discrete Output Point Extended 3 (72_{hex}) – Instance 766...1020

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	DopObj_Value	BYTE	Digital Output (only Bit 0 valid)	-

Common Services

Table 238: Discrete Output Point Extended 2 (6E_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.21 Analog Input Point (67_{hex})

This class enables the reading of data of a particular analog input point (AIP). An analog input point is part of an analog input module.

Instance 0 (Class Attributes)

Table 239: Analog Input Point (67_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 1 ... 255 (Analog input 1 up to 255)

Table 240: Analog Input Point (67_{hex}) – Instance 1 ... 255

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	ARRAY of BYTE	Analog Input	-
2	Get	AipObj_Value_Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 241: Analog Input Point (67_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.22 Analog Input Point Extended 1 (6B_{hex})

The extension of the "Analog Input Point" class enables the reading of data from a fieldbus node that contains over 255 analog outputs (AIPs). The instance scope of the "Analog Input Point Extended 1" class covers AIPs from 256 to 510 in the fieldbus node.

Instance 0 (Class Attributes)

Table 242: Analog Input Point Extended 1 (6B_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 256 ... 510 (Analog Input value 256 up to 510)

Table 243: Analog Input Point Extended 1 (6B_{hex}) – Instance 256 ... 510

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	ARRAY of BYTE	Analog Input	-
2	Get	AipObj_Value_Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 244: Analog Input Point Extended 1 (6B_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.23 Analog Input Point Extended 2 (6F_{hex})

The extension of the "Analog Input Point" class enables the reading of data from a fieldbus node that contains over 510 analog outputs (AIPs). The instance scope of the "Analog Input Point Extended 2" class covers AIPs from 511 to 765 in the fieldbus node.

Instance 0 (Class Attributes)

Table 245: Analog Input Point Extended 2 (6F_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 511 ... 765 (Analog Input 511 up to 765)

Table 246: Analog Input Point Extended 2 (6F_{hex}) – Instance 511 ... 765

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	ARRAY of BYTE	Analog Input	-
2	Get	AipObj_Value_Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 247: Analog Input Point Extended 2 (6F_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.24 Analog Input Point Extended 3 (73_{hex})

The extension of the "Analog Input Point" class enables the reading of data from a fieldbus node that contains over 765 analog outputs (AIPs). The instance scope of the "Analog Input Point Extended 3" class covers AIPs from 766 to 1020 in the fieldbus node.

Instance 0 (Class Attributes)

Table 248: Analog Input Point Extended 3 (73_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 766 ... 1020 (Analog input value 766 up to 1020)

Table 249: Analog Input Point Extended 3 (73_{hex}) – Instance 766 ... 1020

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	ARRAY of BYTE	Analog Input	-
2	Get	AipObj_Value_Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 250: Analog Input Point Extended 3 (73_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.25 Analog Output Point (68_{hex})

This class enables the reading of data of a particular analog output point (AOP). An analog output point is part of an analog output module.

Instance 0 (Class Attributes)

Table 251: Analog Output Point (68_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 1 ... 255 (Analog output value 1 up to 255)

Table 252: Analog Output Point (68_{hex}) – Instance 1...255

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AopObj_Value	ARRAY of BYTE	Analog Output	-
2	Get	AopObj_Value_Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 253: Analog Output Point (68_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.26 Analog Output Point Extended 1 (6C_{hex})

The extension of the "Analog Output Point" class enables the exchange of data from a fieldbus node that contains over 255 analog output points (AOPs). The instance scope of the "Discrete Output Point Extended 1" class covers AOPs from 256 to 510 in the fieldbus node.

Instance 0 (Class Attributes)

Table 254: Analog Output Point Extended 1 (6C_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 256 ... 510 (Analog output value 256 up to 510)

Table 255: Analog Output Point Extended 1 (6C_{hex}) – Instance 256...510

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AopObj_Value	ARRAY of BYTE	Analog Output	-
2	Get	AopObj_Value Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 256: Analog Output Point Extended 1 (6C_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.27 Analog Output Point Extended 2 (70_{hex})

The extension of the "Analog Output Point" class enables the exchange of data from a fieldbus node that contains over 510 analog output points (AOPs). The instance scope of the "Discrete Output Point Extended 2" class covers AOPs from 511 to 765 in the fieldbus node.

Instance 0 (Class Attributes)

Table 257: Analog Output Point Extended 2 (70_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 511 ... 765 (Analog output value 511 up to 765)Table 258: Analog Output Point Extended 2 (70_{hex}) – Instance 511...765

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AopObj_Value	ARRAY of BYTE	Analog Output	-
2	Get	AopObj_Value_Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common ServicesTable 259: Analog Output Point Extended 2 (70_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.28 Analog Output Point Extended 3 (74_{hex})

The extension of the "Analog Output Point" class enables the exchange of data from a fieldbus node that contains over 765 analog output points (AOPs). The instance scope of the "Discrete Output Point Extended 3" class covers AOPs from 766 to 1020 in the fieldbus node.

Instance 0 (Class Attributes)Table 260: Analog Output Point Extended 3 (74_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 766 ... 1020 (Analog output value 766 up to 1020)Table 261: Analog Output Point Extended 3 (74_{hex}) – Instance 766...1020

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AopObj_Value	ARRAY of BYTE	Analog Output	-
2	Get	AopObj_Value_Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 262: Analog Output Point Extended 3 (74_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.29 Module Configuration (80_{hex})

Instance 0 (Class Attributes)

Table 263: Module Configuration (80_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 1 ... 255 (Clamp 0 up to 254)

Table 264: Module Configuration (80_{hex}) – Instance 1...255

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	ModulDescription	WORD	<p>Description of connected modules (module 0 = coupler/controller)</p> <p>Bit 0: Module has inputs Bit 1: Module has outputs Bit 8-14: Data width internally in bit 15: 0/1 Analog/digital module</p> <p>For analog modules, bits 0-14 identify the module type, e.g., 401 for module 750-401</p>	-

Common Services

Table 265: Module Configuration (80_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.30 Module Configuration Extended (81_{hex})

The same as "Module Configuration (80_{hex})" but with a description of module 255.

Instance 0 (Class Attributes)Table 266: Module Configuration Extended (81_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 256 (Clamp 255)Table 267: Module Configuration Extended (81_{hex}) – Instance 256

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	ModulDescription	WORD	Description of connected modules (module 0 = coupler/controller) Bit 0: Module has inputs Bit 1: Module has outputs Bit 8-14: Data width internally in Bit 15: 0/1 Analog/digital module For analog modules, bits 0-14 identify the module type, e.g., 401 for module 750-401	-

Common ServicesTable 268: Module Configuration Extended (81_{hex}) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.31 Input Fieldbus Variable USINT (A0_{hex})

The class enables the reading of data from a particular PLC input variable.

For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for output variables %QB2552...%QB2806.

Instance 0 (Class Attributes)

Table 269: Input fieldbus variable USINT (A0_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x0FF)

Instance 1...255 (Input variable 1 up to 255)

Table 270: Input fieldbus variable USINT (A0_{hex}) – Instance 1...255

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_In_Var	USINT	Fieldbus input variable of the PLC	0

Common Services

Table 271: Input fieldbus variable USINT (A0_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.32 Input Fieldbus Variable USINT Extended 1 (A1_{hex})

The extension of the "Input Fieldbus Variable USINT" class enables the reading of PLC input variable data. The instance scope of the "Input Fieldbus Variable USINT Extended 1" class covers the PLC input variable data from 256 to 510.

For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for input variables %IB2807...%IB3061.

Instance 0 (Class Attributes)Table 272: Input Fieldbus Variable USINT Extended 1 (A1_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x00FF)

Instance 256...510 (Input variable 256 up to 510)Table 273: Input fieldbus variable USINT Extended 1 (A1_{hex}) – Instance 256...510

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_In_Var	USINT	Fieldbus-Input variable of the SPS	0

Common ServicesTable 274: Input fieldbus variable USINT Extended 1 (A1_{hex}) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value.

12.3.5.33 Input Fieldbus Variable USINT Extended 2 (A2_{hex})

The extension of the "Input Fieldbus Variable USINT" class enables the reading of PLC input variable data. The instance scope of the "Input Fieldbus Variable USINT Extended 1" class covers the PLC input variable data from 256 to 510. For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for input variables %IB2807...%IB3061.

Instance 0 (Class Attributes)

Table 275: Input Fieldbus Variable USINT Extended 2 (A2_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	2 (0x0002)

Instance 511...512 (Input variable 511 up to 512)

Table 276: Input Fieldbus Variable USINT Extended 2 (A2_{hex}) – Instance 511...512

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_In_Var	USINT	Fieldbus-Input variable of the SPS	0

Common Services

Table 277: Input fieldbus variable USINT Extended 2 (A2_{hex}) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.34 Output Fieldbus Variable USINT (A3_{hex})

The class enables the exchange of data from a particular PLC output variable.

For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for output variables %QB2552...%QB2806.

Instance 0 (Class Attributes)

Table 278: Output fieldbus variable USINT (A3_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0xFF)

Instance 1...255 (Output variables 1 up to 255)

Table 279: Output fieldbus variable USINT (A3_{hex}) – Instance 1...255

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Fb_Out_Var	USINT	Fieldbus Output variable of the PLC	0

Common Services

Table 280: Output fieldbus variable USINT (A3_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.35 Output Fieldbus Variable USINT Extended 1 (A4_{hex})

The extension of the "Output Fieldbus Variable USINT" class enables the exchange of PLC output variable data. The instance scope of the "Output Fieldbus Variable USINT Extended 1" class covers the PLC output variable data from 256 to 510.

For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for output variables %QB2807...%QB3061.

Instance 0 (Class Attributes)

Table 281: Output Fieldbus variable USINT Extended 1 (A4_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x00FF)

Instance 256...510 (Output variable 256 up to 510)

Table 282: Output Fieldbus Variable USINT Extended 1 (A4_{hex}) – Instance 256...510

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Fb_Out_Var	USINT	Fieldbus output variable of SPS	0

Common Services

Table 283: Output Fieldbus Variable USINT Extended 1 (A4_{hex}) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.36 Output Fieldbus Variable USINT Extended 2 (A5_{hex})

The extension of the "Output Fieldbus Variable USINT" class enables the exchange of PLC output variable data. The instance scope of the "Output Fieldbus Variable USINT Extended 2" class covers the PLC output variable data from 511 to 512.

For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for output variables %QB3062...%QB3063.

Instance 0 (Class Attributes)

Table 284: Output Fieldbus Variable USINT Extended 2 (A5_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	2 (0x0002)

Instance 511...512 (Output variable 511 up to 512)

Table 285: Output Fieldbus Variable USINT Extended 2 (A5_{hex}) – Instance 511...512

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Fb_Out_Var	USINT	Fieldbus-Output variable of SPS	0

Common Services

Tabelle 286: Output Fieldbus Variable USINT Extended 2 (A5_{hex}) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Ja	Ja	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.37 Input Fieldbus Variable UINT (A6_{hex})

This class allows the reading of data from a particular PLC input variable. For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for input variables %IW1276...%IW1530.

Instance 0 (Class Attributes)

Table 287: Input fieldbus variable UINT (A6_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x0FF)

Instance 1...255 (Input variable 1 up to 255)

Table 288: Input fieldbus variable UINT (A6_{hex}) – Instance 1...255

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb In Var	UINT	Fieldbus Input variable of the PLC	0

Common Services

Table 289: Input fieldbus variable UINT (A6_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get Attribute Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set Attribute Single	Modifies an attribute value

12.3.5.38 Input Fieldbus Variable UINT Extended 1 (A7_{hex})

The extension of the "Input Fieldbus Variable UINT" class enables the reading of PLC input variable data. The instance scope of the "Input Fieldbus Variable UINT Extended 1" class covers the PLC input variable data from the PLC input variable 256.

For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for input variable %IW1531.

Instanz 0 (Class Attributes)Table 290: Input Fieldbus Variable UINT Extended 1 (A7_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	1 (0x0001)

Instanz 256 (Input variable 256)Table 291: Input Fieldbus Variable UINT Extended 1 (A7_{hex}) – Instance 256

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_In_Var	UINT	Fieldbus Input variable of the PLC	0

Common ServicesTable 292: Input Fieldbus Variable UINT Extended 1 (A7_{hex}) – Common service

Service code	Service available		Service Name	Description
	Class	Instance		
0E _{hex}	Ja	Ja	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	Nein	Ja	Set_Attribute_Single	Modifies an attribute value

12.3.5.39 Output Fieldbus Variable UINT (A8_{hex})

The class enables the exchange of data from a particular PLC output variable. For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for output variables %QW1276...%QW1530.

Instance 0 (Class Attributes)

Table 293: Output fieldbus variable UINT (A8_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x0FF)

Instance 1...255 (Output variable 1 up to 255)

Table 294: Output fieldbus variable UINT (A8_{hex}) – Instance 1...255

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Fb_Out_Var	UINT	Fieldbus output variable of the PLC	0

Common Services

Table 295: Output fieldbus variable UINT (A8_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.40 Output Fieldbus Variable UINT Extended 1 (A9_{hex})

The extension of the "Output Fieldbus Variable UINT" class enables the exchange of PLC output variable data. The instance scope of the "Output Fieldbus Variable UINT Extended 1" class covers the PLC output variable data from PLC output variables 256.

For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for output variable %QW1531.

Instance 0 (Class Attributes)Table 296: Output Fieldbus Variable UINT Extended 1 (A9_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	1 (0x0001)

Instance 256 (Output variable 256)Table 297: Output Fieldbus Variable UINT Extended 1 (A9_{hex}) – Instance 256

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Fb_Out_Var	UINT	Fieldbus output variable of the SPS	0

Common ServicesTable 298: Output Fieldbus Variable UINT Extended 1 (A9_{hex}) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Ja	Ja	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.41 Input Fieldbus Variable UDINT (AA_{hex})

This class allows the reading of data from a particular PLC input variable. For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for input variables %ID638 ... %ID765.

Instance 0 (Class Attributes)

Table 299: Input fieldbus variable UDINT (AA_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	128 (0x080)

Instance 1...128 (Input variable 1 up to 128)

Table 300: Input fieldbus variable UDINT (AA_{hex}) – Instance 1...128

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_In_Var	UDINT	Fieldbus input variable of the PLC	0

Common Services

Table 301: Input fieldbus variable UDINT (AA_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.42 Input Fieldbus Variable UDINT Offset (AB_{hex})

This class allows the reading of data from a particular PLC input variable. With an offset of 2 bytes to the addresses of the "Input Fieldbus Variable UDINT (AA hex)" class, that means for WAGO-I/O-PRO or CoDeSys the PLC addresses for the input variables %ID638 ... %ID765.



Information

Information about Using the Offset

"Offset of 2 bytes" means:

If instance 1 of this class is read, you obtain High-Word of the address %ID638 and the Low-Word of the address %ID639, etc.

If instance 128 is read, you obtain only the High-Word of the address %ID765.

Instance 0 (Class Attributes)

Table 302: Input Fieldbus Variable UDINT Offset (AB_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	128 (0x080)

Instance 1...128 (Input variable 1 up to 128)

Table 303: Input Fieldbus Variable UDINT Offset (AB_{hex}) – Instance 1...128

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_In_Var	UDINT	Fieldbus-Input variable of the SPS	0

Common Services

Table 304: Input Fieldbus Variable UDINT Offset (AB_{hex}) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 _{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.43 Output Fieldbus Variable UDINT (AC_{hex})

The class enables the exchange of data from a particular PLC output variable. For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for output variables %QD638...%QD765.

Instance 0 (Class Attributes)

Table 305: Input fieldbus variable UDINT (AA_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	128 (0x080)

Instance 1...128 (Output variable 1 up to 128)

Table 306: Input fieldbus variable UDINT (AA_{hex}) – Instance 1...128

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_Out_Var	UDINT	Fieldbus output variable of the PLC	0

Common Services

Table 307: Input fieldbus variable UDINT (AA_{hex}) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.44 Output Fieldbus Variable UDINT Offset (AD_{hex})

The class enables the exchange of data from a particular PLC output variable. With an offset of 2 bytes to the addresses of the "Output Fieldbus Variable UDINT (AC_{hex})" class, that means for WAGO-I/O-PRO or CoDeSys the PLC addresses for %QD638 ... %QD765.



Information

Information about Using the Offset

"Offset of 2 bytes" means:

If instance 1 of this class is read, you obtain High-Word of the address %ID638 and the Low-Word of the address %ID639, etc.

If instance 128 is read, you obtain only the High-Word of the address %ID765..

Instance 0 (Class Attributes)

Table 308: Output Fieldbus Variable UDINT Offset (AD_{hex}) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	128 (0x0080)

Instance 1...128 (Output variable 1 up to 128)

Table 309: Output Fieldbus Variable UDINT Offset (AD_{hex}) – Instance 1...128

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_Out_Var	UDINT	Fieldbus output variable of the SPS	0

Common Services

Table 310: Output Fieldbus Variable UDINT Offset (AD_{hex}) – Common service

Servicecode	Service available		Service-Name	Description
	Casse	Instance		
0E _{hex}	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

13 I/O Modules

13.1 Overview

For modular applications with the WAGO-I/O-SYSTEM 750, different types of I/O modules are available

- Digital Input Modules
- Digital Output Modules
- Analog Input Modules
- Analog Output Modules
- Special Modules
- System Modules

For detailed information on the I/O modules and the module variations, refer to the manuals for the I/O modules.

You will find these manuals on DVD ROM "AUTOMATION Tools and Docs" (order no. 0888-0412) or on the WAGO web pages under <http://www.wago.com>
→ Service → Download → Documentation.



Information

More Information about the WAGO-I/O-SYSTEM

Current information on the modular WAGO-I/O-SYSTEM is available in the Internet under: <http://www.wago.com>

13.2 Process Data Architecture for MODBUS/TCP

With some I/O modules, the structure of the process data is fieldbus specific.

MODBUS/TCP process image uses a word structure (with word alignment). The internal mapping method for data greater than one byte conforms to the Intel format.

The following section describes the process image for various WAGO-I/O-SYSTEM 750 and 753 I/O modules with MODBUS/TCP.

NOTICE

Equipment damage due to incorrect address!

Depending on the specific position of an I/O module in the fieldbus node, the process data of all previous byte or bit-oriented modules must be taken into account to determine its location in the process data map.

The structure of the process data mapping is identical for the PFC process image of the programmable fieldbus controller.

13.2.1 Digital Input Modules

Digital input modules supply one bit of data per channel to specify the signal state for the corresponding channel. These bits are mapped into the Input Process Image.

Some digital modules have an additional diagnostic bit per channel in the Input Process Image. The diagnostic bit is used for detecting faults that occur (e.g., wire breaks and/or short circuits).

When analog input modules are also present in the node, the digital data is always appended after the analog data in the Input Process Image, grouped into bytes.

13.2.1.1 1 Channel Digital Input Module with Diagnostics

750-435

Table 311: 1 Channel Digital Input Module with Diagnostics

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 1	Data bit DI 1

13.2.1.2 2 Channel Digital Input Modules

750-400, -401, -405, -406, -410, -411, -412, -427, -438, (and all variations),
753-400, -401, -405, -406, -410, -411, -412, -427

Table 312: 2 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.3 2 Channel Digital Input Module with Diagnostics

750-419, -421, -424, -425,
753-421, -424, -425

Table 313: 2 Channel Digital Input Module with Diagnostics

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.4 2 Channel Digital Input Module with Diagnostics and Output Process Data

750-418,
753-418

The digital input module supplies a diagnostic and acknowledge bit for each input channel. If a fault condition occurs, the diagnostic bit is set. After the fault condition is cleared, an acknowledge bit must be set to re-activate the input. The diagnostic data and input data bit is mapped in the Input Process Image, while the acknowledge bit is in the Output Process Image.

Table 314: 2 Channel Digital Input Module with Diagnostics and Output Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Acknowledgement bit Q 2 Channel 2	Acknowledgement bit Q 1 Channel 1	0	0

13.2.1.5 4 Channel Digital Input Modules

750-402, -403, -408, -409, -414, -415, -422, -423, -428, -432, -433, -1420, -1421, -1422
753-402, -403, -408, -409, -415, -422, -423, -428, -432, -433, -440

Table 315: 4 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.6 8 Channel Digital Input Modules

750-430, -431, -436, -437, -1415, -1416, -1417
753-430, -431, -434

Table 316: 8 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.7 16 Channel Digital Input Modules

750-1400, -1402, -1405, -1406, -1407

Table 317: 16 Channel Digital Input Modules

Input Process Image															
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit	Data bit
DI 16	DI 15	DI 14	DI 13	DI 12	DI 11	DI 10	DI 9	8	DI 7	DI 6	DI 5	DI 4	DI 3	DI 2	DI 1
Chan- nel 16	Chan- nel 15	Chan- nel 14	Chan- nel 13	Chan- nel 12	Chan- nel 11	Chan- nel 10	Chan- nel 9	Chan- nel 8	Chan- nel 7	Chan- nel 6	Chan- nel 5	Chan- nel 4	Chan- nel 3	Chan- nel 2	Chan- nel 1

13.2.2 Digital Output Modules

Digital output modules use one bit of data per channel to control the output of the corresponding channel. These bits are mapped into the Output Process Image.

Some digital modules have an additional diagnostic bit per channel in the Input Process Image. The diagnostic bit is used for detecting faults that occur (e.g., wire breaks and/or short circuits). For modules with diagnostic bit is set, also the data bits have to be evaluated.

When analog output modules are also present in the node, the digital image data is always appended after the analog data in the Output Process Image, grouped into bytes.

13.2.2.1 1 Channel Digital Output Module with Input Process Data

750-523

The digital output modules deliver 1 bit via a process value Bit in the output process image, which is illustrated in the input process image. This status image shows "manual mode".

Table 318: 1 Channel Digital Output Module with Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						not used	Status bit "Manual Operation"

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						not used	controls DO 1 Channel 1

13.2.2.2 2 Channel Digital Output Modules

750-501, -502, -509, -512, -513, -514, -517, -535, (and all variations),
753-501, -502, -509, -512, -513, -514, -517

Table 319: 2 Channel Digital Output Modules

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.3 2 Channel Digital Input Modules with Diagnostics and Input Process Data

750-507 (-508), -522,
753-507

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 320: 2 Channel Digital Input Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						controls DO 2 Channel 2	controls DO 1 Channel 1

750-506,
753-506

The digital output module has 2-bits of diagnostic information for each output channel. The 2-bit diagnostic information can then be decoded to determine the exact fault condition of the module (i.e., overload, a short circuit, or a broken wire). The 4-bits of diagnostic data are mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 321: 2 Channel Digital Input Modules with Diagnostics and Input Process Data 75x-506

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 3 Channel 2	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Diagnostic bit S 0 Channel 1

Diagnostic bits S1/S0, S3/S2: = '00' standard mode
 Diagnostic bits S1/S0, S3/S2: = '01' no connected load/short circuit against +24 V
 Diagnostic bits S1/S0, S3/S2: = '10' Short circuit to ground/overload

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				not used	not used	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.4 4 Channel Digital Output Modules

750-504, -516, -519, -531,
753-504, -516, -531, -540

Table 322: 4 Channel Digital Output Modules

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.5 4 Channel Digital Output Modules with Diagnostics and Input Process Data

750-532

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 323: 4 Channel Digital Output Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 4 Channel 4	Diagnostic bit S 3 Channel 3	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Diagnostic bit S = '0' no Error

Diagnostic bit S = '1' overload, short circuit, or broken wire

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.6 8 Channel Digital Output Module

750-530, -536, -1515, -1516
753-530, -534

Table 324: 8 Channel Digital Output Module

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.7 8 Channel Digital Output Modules with Diagnostics and Input Process Data

750-537

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 325: 8 Channel Digital Output Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Diagnostic bit S 8 Channel 8	Diagnostic bit S 7 Channel 7	Diagnostic bit S 6 Channel 6	Diagnostic bit S 5 Channel 5	Diagnostic bit S 4 Channel 4	Diagnostic bit S 3 Channel 3	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Diagnostic bit S = '0' no Error
Diagnostic bit S = '1' overload, short circuit, or broken wire

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.8 16 Channel Digital Output Modules

750-1500, -1501, -1504, -1505

Table 326: 16 Channel Digital Output Modules

Output Process Image															
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 16 Channel 16	controls DO 15 Channel 15	controls DO 14 Channel 14	controls DO 13 Channel 13	controls DO 12 Channel 12	controls DO 11 Channel 11	controls DO 10 Channel 10	controls DO 9 Channel 9	controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.9 8 Channel Digital Input/Output Modules

750-1502, -1506

Table 327: 8 Channel Digital Input/Output Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.3 Analog Input Modules

The hardware of an analog input module has 16 bits of measured analog data per channel and 8 bits of control/status.

However, the coupler/controller with MODBUS/TCP does not have access to the 8 control/status bits.

Therefore, the coupler/controller with MODBUS/TCP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Input Process Image.

When digital input modules are also present in the node, the analog input data is always mapped into the Input Process Image in front of the digital data.

Information



Information to the structure of the Control/Status byte

For detailed information about the structure of a particular module's control/status byte, please refer to that module's manual. Manuals for each module can be found on the Internet under: <http://www.wago.com>.

13.2.3.1 1 Channel Analog Input Modules

750-491, (and all variations)

Table 328: 1 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Measured Value U_D
1	D3	D2	Measured Value U_{ref}

13.2.3.2 2 Channel Analog Input Modules

750-452, -454, -456, -461, -462, -465, -466, -467, -469, -472, -474, -475, 476, -477, -478, -479, -480, -481, -483, -485, -492, (and all variations),
753-452, -454, -456, -461, -465, -466, -467, -469, -472, -474, -475, 476, -477, 478, -479, -483, -492, (and all variations)

Table 329: 2 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Measured Value Channel 1
1	D3	D2	Measured Value Channel 2

13.2.3.3 4 Channel Analog Input Modules

750-453, -455, -457, -459, -460, -468, (and all variations),
753-453, -455, -457, -459

Table 330: 4 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Measured Value Channel 1
1	D3	D2	Measured Value Channel 2
2	D5	D4	Measured Value Channel 3
3	D7	D6	Measured Value Channel 4

13.2.4 Analog Output Modules

The hardware of an analog output module has 16 bits of measured analog data per channel and 8 bits of control/status. However, the coupler/controller with MODBUS/TCP does not have access to the 8 control/status bits. Therefore, the coupler/controller with MODBUS/TCP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Output Process Image.

When digital output modules are also present in the node, the analog output data is always mapped into the Output Process Image in front of the digital data.

Information



Information to the structure of the Control/Status byte

For detailed information about the structure of a particular module's control/status byte, please refer to that module's manual. Manuals for each module can be found on the Internet under: <http://www.wago.com>.

13.2.4.1 2 Channel Analog Output Modules

750-550, -552, -554, -556, -560, -562, 563, -585, (and all variations),
753-550, -552, -554, -556

Table 331: 2 Channel Analog Output Modules

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Output Value Channel 1
1	D3	D2	Output Value Channel 2

13.2.4.2 4 Channel Analog Output Modules

750-553, -555, -557, -559,
753-553, -555, -557, -559

Table 332: 4 Channel Analog Output Modules

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Output Value Channel 1
1	D3	D2	Output Value Channel 2
2	D5	D4	Output Value Channel 3
3	D7	D6	Output Value Channel 4

13.2.5 Specialty Modules

WAGO has a host of Specialty I/O modules that perform various functions. With individual modules beside the data bytes also the control/status byte is mapped in the process image.

The control/status byte is required for the bidirectional data exchange of the module with the higher-ranking control system. The control byte is transmitted from the control system to the module and the status byte from the module to the control system.

This allows, for example, setting of a counter with the control byte or displaying of overshooting or undershooting of the range with the status byte.

The control/status byte always is in the process image in the Low byte.

Information



Information to the structure of the Control/Status byte

For detailed information about the structure of a particular module's control/status byte, please refer to that module's manual. Manuals for each module can be found on the Internet under: <http://www.wago.com>.

13.2.5.1 Counter Modules

750-404, (and all variations except of /000-005),
753-404, (and variation /000-003)

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/status). The counter value is supplied as 32 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

Table 333: Counter Modules 750-404, (and all variations except of /000-005),
753-404, (and variation /000-003)

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	S	Status byte
1	D1	D0	Counter value
2	D3	D2	

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C	Control byte
1	D1	D0	Counter setting value
2	D3	D2	

750-404/000-005

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/ status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

Table 334: Counter Modules 750-404/000-005

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	S	Status byte
1	D1	D0	Counter Value of Counter 1
2	D3	D2	Counter Value of Counter 2

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C	Control byte
1	D1	D0	Counter Setting Value of Counter 1
2	D3	D2	Counter Setting Value of Counter 2

750-638,
753-638

The above Counter Modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 2 bytes of control/status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

Table 335: Counter Modules 750-638, 753-638

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	S0	Status byte von Counter 1
1	D1	D0	Counter Value von Counter 1
2	-	S1	Status byte von Counter 2
3	D3	D2	Counter Value von Counter 2

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C0	Control byte von Counter 1
1	D1	D0	Counter Setting Value von Counter 1
2	-	C1	Control byte von Counter 2
3	D3	D2	Counter Setting Value von Counter 2

13.2.5.2 Pulse Width Modules

750-511, (and all variations /xxx-xxx)

The above Pulse Width modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of channel data and 2 bytes of control/status). The two channel values are supplied as 16 bits. Each channel has its own control/status byte. The following table illustrates the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

Table 336: Pulse Width Modules 750-511, /xxx-xxx

Input and Output Process			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C0/S0	Control/Status byte of Channel 1
1	D1	D0	Data Value of Channel 1
2	-	C1/S1	Control/Status byte of Channel 2
3	D3	D2	Data Value of Channel 2

13.2.5.3 Serial Interface Modules with alternative Data Format

750-650, (and the variations /000-002, -004, -006, -009, -010, -011, -012, -013),
750-651, (and the variations /000-001, -002, -003),
750-653, (and the variations /000-002, -007),
753-650, -653

Note



The process image of the / 003-000-variants depends on the parameterized operating mode!

With the freely parametrizable variations /003 000 of the serial interface modules, the desired operation mode can be set. Dependent on it, the process image of these modules is then the same, as from the appropriate variation.

The above Serial Interface Modules with alternative data format have a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 2 words mapped into each image. Word alignment is applied.

Table 337: Serial Interface Modules with alternative Data Format

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	C/S	Data byte	Control/status byte
1	D2	D1	Data bytes	

13.2.5.4 Serial Interface Modules with Standard Data Format

750-650/000-001, -014, -015, -016
750-653/000-001, -006

The above Serial Interface Modules with Standard Data Format have a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 3 words mapped into each image. Word alignment is applied.

Table 338: Serial Interface Modules with Standard Data Format

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	C/S	Data byte	Control/status byte
1	D2	D1	Data bytes	
2	D4	D3		

13.2.5.5 Data Exchange Module

750-654, (and the variation /000-001)

The Data Exchange modules have a total of 4 bytes of user data in both the Input and Output Process Image. The following tables illustrate the Input and Output Process Image, which has a total of 2 words mapped into each image. Word alignment is applied.

Table 339: Data Exchange Module

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D1	D0	Data bytes	
1	D3	D2		

13.2.5.6 SSI Transmitter Interface Modules

750-630 (and all variations)



Note

The process image of the / 003-000-variants depends on the parameterized operating mode!

The operating mode of the configurable /003-000 I/O module versions can be set. Based on the operating mode, the process image of these I/O modules is then the same as that of the respective version.

The above SSI Transmitter Interface modules have a total of 4 bytes of user data in the Input Process Image, which has 2 words mapped into the image. Word alignment is applied.

Table 340: SSI Transmitter Interface Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Data bytes
1	D3	D2	

13.2.5.7 Incremental Encoder Interface Modules

750-631/000-004, -010, -011

The above Incremental Encoder Interface modules have 5 bytes of input data and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which have 4 words into each image. Word alignment is applied.

Table 341: Incremental Encoder Interface Modules 750-631/000-004, --010, -011

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	S	not used Status byte
1	D1	D0	Counter word
2	-	-	not used
3	D4	D3	Latch word

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C	not used Control byte
1	D1	D0	Counter setting word
2	-	-	not used
3	-	-	not used

750-634

The above Incremental Encoder Interface module has 5 bytes of input data (6 bytes in cycle duration measurement mode) and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which has 4 words mapped into each image. Word alignment is applied.

Table 342: Incremental Encoder Interface Modules 750-634

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	S	not used Status byte
1	D1	D0	Counter word
2	-	(D2) *)	not used (Periodic time)
3	D4	D3	Latch word

*) If cycle duration measurement mode is enabled in the control byte, the cycle duration is given as a 24-bit value that is stored in D2 together with D3/D4.

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C	not used Control byte
1	D1	D0	Counter setting word
2	-	-	not used
3	-	-	

750-637

The above Incremental Encoder Interface Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of encoder data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Table 343: Incremental Encoder Interface Modules 750-637

Input and Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C0/S0	Control/Status byte of Channel 1
1	D1	D0	Data Value of Channel 1
2	-	C1/S1	Control/Status byte of Channel 2
3	D3	D2	Data Value of Channel 2

750-635,
753-635

The above Digital Pulse Interface module has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

Table 344: Digital Pulse Interface Modules 750-635

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	C0/S0	Data byte	Control/status byte
1	D2	D1	Data bytes	

13.2.5.8 DC-Drive Controller

750-636

The DC-Drive Controller maps 6 bytes into both the input and output process image. The data sent and received are stored in up to 4 input and output bytes (D0 ... D3). Two control bytes (C0, C1) and two status bytes (S0/S1) are used to control the I/O module and the drive.

In addition to the position data in the input process image (D0 ... D3), it is possible to display extended status information (S2 ... S5). Then the three control bytes (C1 ... C3) and status bytes (S1 ... S3) are used to control the data flow.

Bit 3 of control byte C1 (C1.3) is used to switch between the process data and the extended status bytes in the input process image (Extended Info_ON). Bit 3 of status byte S1 (S1.3) is used to acknowledge the switching process.

Table 345: DC-Drive Controller 750-636

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	S1	S0	Status byte S1	Status byte S0
1	D1*) / S3**)	D0*) / S2**)	Actual position*) / Extended status byte S3**)	Actual position (LSB) / Extended status byte S2**)
2	D3*) / S5**)	D2*) / S4**)	Actual position (MSB) / Extended status byte S3**)	Actual position*) / Extended status byte S4**)

*) ExtendedInfo_ON = '0'.

***) ExtendedInfo_ON = '1'.

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	C1	C0	Control byte C1	Control byte C0
1	D1	D0	Setpoint position	Setpoint position (LSB)
2	D3	D2	Setpoint position (MSB)	Setpoint position

13.2.5.9 Stepper Controller

750-670

The Stepper controller RS422 / 24 V / 20 mA 750-670 provides the fieldbus coupler 12 bytes input and output process image via 1 logical channel. The data to be sent and received are stored in up to 7 output bytes (D0 ... D6) and 7 input bytes (D0 ... D6), depending on the operating mode.

Output byte D0 and input byte D0 are reserved and have no function assigned.

One I/O module control and status byte (C0, S0) and 3 application control and status bytes (C1 ... C3, S1 ... S3) provide the control of the data flow.

Switching between the two process images is conducted through bit 5 in the control byte (C0 (C0.5)). Activation of the mailbox is acknowledged by bit 5 of the status byte S0 (S0.5).

Table 346: Stepper Controller RS 422 / 24 V / 20 mA 750-670

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	reserved	S0	reserved	Status byte S0
1	D1	D0	Process data*) / Mailbox**)	
2	D3	D2		
3	D5	D4		
4	S3	D6	Status byte S3	Process data*) / reserved**)
5	S1	S2	Status byte S1	Status byte S2

*) Cyclic process image (Mailbox disabled)

***) Mailbox process image (Mailbox activated)

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	reserved	C0	reserved	Control byte C0
1	D1	D0	Process data*) / Mailbox**)	
2	D3	D2		
3	D5	D4		
4	C3	D6	Control byte C3	Process data*) / reserved**)
5	C1	C2	Control byte C1	Control byte C2

*) Cyclic process image (Mailbox disabled)

***) Mailbox process image (Mailbox activated)

13.2.5.10 RTC Module

750-640

The RTC Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of module data and 1 byte of control/status and 1 byte ID for command). The following table illustrates the Input and Output Process Image, which have 3 words mapped into each image. Word alignment is applied.

Table 347: RTC Module 750-640

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	ID	C/S	Command byte	Control/status byte
1	D1	D0	Data bytes	
2	D3	D2		

13.2.5.11 DALI/DSI Master Module

750-641

The DALI/DSI Master module has a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of module data and 1 byte of control/status). The following tables illustrate the Input and Output Process Image, which have 3 words mapped into each image. Word alignment is applied.

Table 348: DALI/DSI Master module 750-641

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	S	DALI Response	Status byte
1	D2	D1	Message 3	DALI Address
2	D4	D3	Message 1	Message 2

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	C	DALI command, DSI dimming value	Control byte
1	D2	D1	Parameter 2	DALI Address
2	D4	D3	Command extension	Parameter 1

13.2.5.12 EnOcean Radio Receiver

750-642

The EnOcean radio receiver has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following tables illustrate the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

Table 349: EnOcean Radio Receiver 750-642

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	S	Data byte	Status byte
1	D2	D1	Data bytes	

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C	not used	Control byte
1	-	-	not used	

13.2.5.13 MP Bus Master Module

750-643

The MP Bus Master Module has a total of 8 bytes of user data in both the Input and Output Process Image (6 bytes of module data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Table 350: MP Bus Master Module 750-643

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	C1/S1	C0/S0	extended Control/ Status byte	Control/status byte
1	D1	D0	Data bytes	
2	D3	D2		
3	D5	D4		

13.2.5.14 Bluetooth® RF-Transceiver

750-644

The size of the process image for the *Bluetooth*® module can be adjusted to 12, 24 or 48 bytes.

It consists of a control byte (input) or status byte (output); an empty byte; an overlayable mailbox with a size of 6, 12 or 18 bytes (mode 2); and the *Bluetooth*® process data with a size of 4 to 46 bytes.

Thus, each *Bluetooth*® module uses between 12 and 48 bytes in the process image. The sizes of the input and output process images are always the same.

The first byte contains the control/status byte; the second contains an empty byte.

Process data attach to this directly when the mailbox is hidden. When the mailbox is visible, the first 6, 12 or 18 bytes of process data are overlaid by the mailbox data, depending on their size. Bytes in the area behind the optionally visible mailbox contain basic process data. The internal structure of the *Bluetooth*® process data can be found in the documentation for the *Bluetooth*® 750-644 RF Transceiver.

The mailbox and the process image sizes are set with the startup tool WAGO-I/O-CHECK.

Table 351: Bluetooth® RF-Transceiver 750-644

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C0/S0	not used	Control/status byte
1	D1	D0	Mailbox (0, 3, 6 or 9 words) and Process data (2-23 words)	
2	D3	D2		
3	D5	D4		
...		
max. 23	D45	D44		

13.2.5.15 Vibration Velocity/Bearing Condition Monitoring VIB I/O

750-645

The Vibration Velocity/Bearing Condition Monitoring VIB I/O has a total of 12 bytes of user data in both the Input and Output Process Image (8 bytes of module data and 4 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 8 words mapped into each image. Word alignment is applied.

Table 352: Vibration Velocity/Bearing Condition Monitoring VIB I/O 750-645

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C0/S0	not used	Control/status byte (log. Channel 1, Sensor input 1)
1	D1	D0	Data bytes (log. Channel 1, Sensor input 1)	
2	-	C1/S1	not used	Control/status byte (log. Channel 2, Sensor input 2)
3	D3	D2	Data bytes (log. Channel 2, Sensor input 2)	
4	-	C2/S2	not used	Control/status byte (log. Channel 3, Sensor input 1)
5	D5	D4	Data bytes (log. Channel 3, Sensor input 3)	
6	-	C3/S3	not used	Control/status byte (log. Channel 4, Sensor input 2)
7	D7	D6	Data bytes (log. Channel 4, Sensor input 2)	

13.2.5.16 KNX/EIB/TP1 Module

753-646

The KNX/TP1 module appears in router and device mode with a total of 24-byte user data within the input and output area of the process image, 20 data bytes and 2 control/status bytes. Even though the additional bytes S1 or C1 are transferred as data bytes, they are used as extended status and control bytes. The opcode is used for the read/write command of data and the triggering of specific functions of the KNX/EIB/TP1 module. Word-alignment is used to assign 12 words in the process image. Access to the process image is not possible in router mode. Telegrams can only be tunneled.

In device mode, access to the KNX data can only be performed via special function blocks of the IEC application. Configuration using the ETS engineering tool software is required for KNX.

Table 353: KNX/EIB/TP1 Module 753-646

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	S0	not used	Status byte
1	S1	OP	extended Status byte	Opcode
2	D1	D0	Data byte 1	Data byte 0
3	D3	D2	Data byte 3	Data byte 2
4	D5	D4	Data byte 5	Data byte 4
5	D7	D6	Data byte 7	Data byte 6
6	D9	D8	Data byte 9	Data byte 8
7	D11	D10	Data byte 11	Data byte 10
8	D13	D12	Data byte 13	Data byte 12
9	D15	D14	Data byte 15	Data byte 14
10	D17	D16	Data byte 17	Data byte 16
11	D19	D18	Data byte 19	Data byte 18

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C0	not used	Control byte
1	C1	OP	extended Control byte	Opcode
2	D1	D0	Data byte 1	Data byte 0
3	D3	D2	Data byte 3	Data byte 2
4	D5	D4	Data byte 5	Data byte 4
5	D7	D6	Data byte 7	Data byte 6
6	D9	D8	Data byte 9	Data byte 8
7	D11	D10	Data byte 11	Data byte 10
8	D13	D12	Data byte 13	Data byte 12
9	D15	D14	Data byte 15	Data byte 14
10	D17	D16	Data byte 17	Data byte 16
11	D19	D18	Data byte 19	Data byte 18

13.2.5.17 AS-interface Master Module

750-655

The length of the process image of the AS-interface master module can be set to fixed sizes of 12, 20, 24, 32, 40 or 48 bytes.

It consists of a control or status byte, a mailbox with a size of 0, 6, 10, 12 or 18 bytes and the AS-interface process data, which can range from 0 to 32 bytes.

The AS-interface master module has a total of 6 to maximally 24 words data in both the Input and Output Process Image. Word alignment is applied.

The first Input and output word, which is assigned to an AS-interface master module, contains the status / control byte and one empty byte.

Subsequently the mailbox data are mapped, when the mailbox is permanently superimposed (Mode 1).

In the operating mode with suppressible mailbox (Mode 2), the mailbox and the cyclical process data are mapped next.
The following words contain the remaining process data.

The mailbox and the process image sizes are set with the startup tool WAGO-I/O-CHECK.

Table 354: AS-interface Master module 750-655

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C0/S0	not used	Control/status byte
1	D1	D0	Mailbox (0, 3, 5, 6 or 9 words)/ Process data (0-16 words)	
2	D3	D2		
3	D5	D4		
...		
max. 23	D45	D44		

13.2.6 System Modules

13.2.6.1 System Modules with Diagnostics

750-610, -611

The modules provide 2 bits of diagnostics in the Input Process Image for monitoring of the internal power supply.

Table 355: System Modules with Diagnostics 750-610, -611

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 2 Fuse	Diagnostic bit S 1 Fuse

13.2.6.2 Binary Space Module

750-622

The Binary Space Modules behave alternatively like 2 channel digital input modules or output modules and seize depending upon the selected settings 1, 2, 3 or 4 bits per channel. According to this, 2, 4, 6 or 8 bits are occupied then either in the process input or the process output image.

Table 356: Binary Space Module 750-622 (with behavior like 2 channel digital input)

Input and Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(Data bit DI 8)	(Data bit DI 7)	(Data bit DI 6)	(Data bit DI 5)	(Data bit DI 4)	(Data bit DI 3)	Data bit DI 2	Data bit DI 1

13.3 Process Data Architecture for EtherNet/IP

With some I/O modules, the structure of the process data is fieldbus specific.

In the case of a fieldbus controller with EtherNet/IP, the process image uses a word structure (with word alignment). The internal mapping method for data greater than one byte conforms to the Intel format.

The following section describes the process image for various WAGO-I/O-SYSTEM 750 and 753 I/O modules when using a fieldbus controller with EtherNet/IP.

For the PFC process image of the programmable fieldbus controller is the structure of the process data mapping identical.

NOTICE

Equipment damage due to incorrect address!

Depending on the specific position of an I/O module in the fieldbus node, the process data of all previous byte or bit-oriented modules must be taken into account to determine its location in the process data map.

13.3.1 Digital Input Modules

Digital input modules supply one bit of data per channel to specify the signal state for the corresponding channel. These bits are mapped into the Input Process Image.

Some digital I/O modules have an additional diagnostic bit per channel in the input process image. The diagnostic bit detects faults (e.g., wire breakage, overloads and/or short circuits). For some I/O modules, the data bits also have to be evaluated with the set diagnostic bit.

When analog input modules are also present in the node, the digital data is always appended after the analog data in the Input Process Image, grouped into bytes.

1 sub index is assigned for each 8 bit.

Each input channel seizes one Instance in the Discrete Input Point Object (Class 0x65).

13.3.1.1 1 Channel Digital Input Module with Diagnostics

750-435

Table 357: 1 Channel Digital Input Module with Diagnostics

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 1	Data bit DI 1

The input modules seize 2 Instances in Class (0x65).

13.3.1.2 2 Channel Digital Input Modules

750-400, -401, -405, -406, -410, -411, -412, -427, -438, (and all variations),
753-400, -401, -405, -406, -410, -411, -412, -427

Table 358: 2 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

The input modules seize 2 Instances in Class (0x65).

13.3.1.3 2 Channel Digital Input Module with Diagnostics

750-419, -421, -424, -425,
753-421, -424, -425

Table 359: 2 Channel Digital Input Module with Diagnostics

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

The input modules seize 4 Instances in Class (0x65).

13.3.1.4 2 Channel Digital Input Module with Diagnostics and Output Process Data

750-418,
753-418

The digital input module supplies a diagnostic and acknowledge bit for each input channel. If a fault condition occurs, the diagnostic bit is set. After the fault condition is cleared, an acknowledge bit must be set to re-activate the input. The diagnostic data and input data bit is mapped in the Input Process Image, while the acknowledge bit is in the Output Process Image.

Table 360: 2 Channel Digital Input Module with Diagnostics and Output Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

The input modules seize 4 Instances in Class (0x65).

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Acknowledgement bit Q 2 Channel 2	Acknowledgement bit Q 1 Channel 1	0	0

And the input modules seize 4 Instances in Class (0x66).

13.3.1.5 4 Channel Digital Input Modules

750-402, -403, -408, -409, -414, -415, -422, -423, -428, -432, -433, -1420, -1421, -1422

753-402, -403, -408, -409, -415, -422, -423, -428, -432, -433, -440

Table 361: 4 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

The input modules seize 4 Instances in Class (0x65).

13.3.1.6 8 Channel Digital Input Modules

750-430, -431, -436, -437, -1415, -1416, -1417

753-430, -431, -434

Table 362: 8 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

The input modules seize 8 Instances in Class (0x65).

13.3.1.7 16 Channel Digital Input Modules

750-1400, -1402, -1405, -1406, -1407

Table 363: 16 Channel Digital Input Modules

Input Process Image															
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 16 Channel 16	Data bit DI 15 Channel 15	Data bit DI 14 Channel 14	Data bit DI 13 Channel 13	Data bit DI 12 Channel 12	Data bit DI 11 Channel 11	Data bit DI 10 Channel 10	Data bit DI 9 Channel 9	Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.3.2 Digital Output Modules

Digital output modules use one bit of data per channel to control the output of the corresponding channel. These bits are mapped into the Output Process Image.

Some digital modules have an additional diagnostic bit per channel in the Input Process Image. The diagnostic bit is used for detecting faults that occur (e.g., wire breaks and/or short circuits). With some I/O modules, with set diagnostic bit, additionally the data bits must be evaluated.

When analog output modules are also present in the node, the digital image data is always appended after the analog data in the Output Process Image, grouped into bytes.

For each 8 bits a subindex is occupied.

Each output channel occupies one instance in the Discrete Output Point Object (Class 0x 66).

13.3.2.1 1 Channel Digital Output Module with Input Process Data

750-523

The digital output modules deliver 1 bit via a process value Bit in the output process image, which is illustrated in the input process image. This status image shows "manual mode".

Table 364: 1 Channel Digital Output Module with Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						not used	Status bit "Manual Operation"

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						not used	controls DO 1 Channel 1

And the output modules seize 2 Instances in Class (0x66).

13.3.2.2 2 Channel Digital Output Modules

750-501, -502, -509, -512, -513, -514, -517, -535, (and all variations),
753-501, -502, -509, -512, -513, -514, -517

Table 365: 2 Channel Digital Output Modules

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						controls DO 2 Channel 2	controls DO 1 Channel 1

The output modules seize 2 Instances in Class (0x66).

13.3.2.3 2 Channel Digital Input Modules with Diagnostics and Input Process Data

750-507 (-508), -522,
753-507

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 366: 2 Channel Digital Input Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						controls DO 2 Channel 2	controls DO 1 Channel 1

And the output modules seize 2 Instances in Class (0x66).

750-506,
753-506

The digital output module has 2-bits of diagnostic information for each output channel. The 2-bit diagnostic information can then be decoded to determine the exact fault condition of the module (i.e., overload, a short circuit, or a broken wire). The 4-bits of diagnostic data are mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 367: 2 Channel Digital Input Modules with Diagnostics and Input Process Data 75x-506

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 3 Channel 2	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Diagnostic bit S 0 Channel 1

Diagnostic bits S1/S0, S3/S2: = '00' standard mode
 Diagnostic bits S1/S0, S3/S2: = '01' no connected load/short circuit against +24 V
 Diagnostic bits S1/S0, S3/S2: = '10' Short circuit to ground/overload
 The output modules seize 4 Instances in Class (0x65).

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				not used	not used	controls DO 2 Channel 2	controls DO 1 Channel 1

And the output modules seize 4 Instances in Class (0x66).

13.3.2.4 4 Channel Digital Output Modules

750-504, -516, -519, -531,
753-504, -516, -531, -540

Table 368: 4 Channel Digital Output Modules

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

The output modules seize 4 Instances in Class (0x66).

13.3.2.5 4 Channel Digital Output Modules with Diagnostics and Input Process Data

750-532

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 369: 4 Channel Digital Output Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 4 Channel 4	Diagnostic bit S 3 Channel 3	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Diagnostic bit S = '0' no Error
 Diagnostic bit S = '1' overload, short circuit, or broken wire
 The output modules seize 4 Instances in Class (0x65).

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

And the output modules seize 4 Instances in Class (0x66).

13.3.2.6 8 Channel Digital Output Module

750-530, -536, -1515, -1516

753-530, -534

Table 370: 8 Channel Digital Output Module

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

The output modules seize 8 Instances in Class (0x66).

13.3.2.7 8 Channel Digital Output Modules with Diagnostics and Input Process Data

750-537

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 371: 8 Channel Digital Output Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Diagnostic bit S 8 Channel 8	Diagnostic bit S 7 Channel 7	Diagnostic bit S 6 Channel 6	Diagnostic bit S 5 Channel 5	Diagnostic bit S 4 Channel 4	Diagnostic bit S 3 Channel 3	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Diagnostic bit S = '0' no Error

Diagnostic bit S = '1' overload, short circuit, or broken wire

The output modules seize 8 Instances in Class (0x65).

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

And the output modules seize 8 Instances in Class (0x66).

13.3.2.8 16 Channel Digital Output Modules

750-1500, -1501, -1504, -1505

Table 372: 16 Channel Digital Output Modules

Output Process Image															
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 16 Channel 16	controls DO 15 Channel 15	controls DO 14 Channel 14	controls DO 13 Channel 13	controls DO 12 Channel 12	controls DO 11 Channel 11	controls DO 10 Channel 10	controls DO 9 Channel 9	controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

The output modules seize 16 Instances in Class (0x66).

13.3.2.9 8 Channel Digital Input/Output Modules

750-1502, -1506

Table 373: 8 Channel Digital Input/Output Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

The input/output modules seize 8 Instances in Class (0x65).

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

The input/output modules seize 8 Instances in Class (0x66).

13.3.3 Analog Input Modules

The hardware of an analog input module has 16 bits of measured analog data per channel and 8 bits of control/status.

However, the coupler/controller with EtherNet/IP does not have access to the 8 control/status bits.

Therefore, the coupler/controller with MODBUS/TCP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Input Process Image.

When digital input modules are also present in the node, the analog input data is always mapped into the Input Process Image in front of the digital data.

Each input channel seizes one Instance in the Analog Input Point Object (Class 0x67).



Note

Information for the control/status byte development

Please refer to the corresponding description of the I/O modules for the structure of the control/status bytes. You can find a manual with the relevant I/O module description on the WAGO home page: at: <http://www.wago.com>.

13.3.3.1 1 Channel Analog Input Modules

750-491, (and all variations)

Table 374: 1 Channel Analog Input Modules

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	D1	D0	Measured Value U_D
n+1	D3	D2	Measured Value U_{ref}

The input modules represent 2x2 bytes and seize 2 Instances in Class (0x67).

13.3.3.2 2 Channel Analog Input Modules

750-452, -454, -456, -461, -462, -465, -466, -467, -469, -472, -474, -475, 476, -477, -478, -479, -480, -481, -483, -485, -492, (and all variations),
753-452, -454, -456, -461, -465, -466, -467, -469, -472, -474, -475, 476, -477, 478, -479, -483, -492, (and all variations)

Table 375: 2 Channel Analog Input Modules

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	D1	D0	Measured Value Channel 1
n+1	D3	D2	Measured Value Channel 2

The input modules represent 2x2 bytes and seize 2 Instances in Class (0x67).

13.3.3.3 4 Channel Analog Input Modules

750-453, -455, -457, -459, -460, -468, (and all variations),
753-453, -455, -457, -459

Table 376: 4 Channel Analog Input Modules

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	D1	D0	Measured Value Channel 1
n+1	D3	D2	Measured Value Channel 2
n+2	D5	D4	Measured Value Channel 3
n+3	D7	D6	Measured Value Channel 4

The input modules represent 4x2 bytes and seize 4 Instances in Class (0x67).

13.3.4 Analog Output Modules

The hardware of an analog output module has 16 bits of measured analog data per channel and 8 bits of control/status. However, the coupler/controller with EtherNet/IP does not have access to the 8 control/status bits. Therefore, the coupler/controller with EtherNet/IP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Output Process Image.

When digital output modules are also present in the node, the analog output data is always mapped into the Output Process Image in front of the digital data.

Each output channel seizes one Instance in the Analog Output Point Object (Class 0x68).



Information

Information to the structure of the Control/Status byte

For detailed information about the structure of a particular module's control/status byte, please refer to that module's manual. Manuals for each module can be found on the Internet under: <http://www.wago.com>.

13.3.4.1 2 Channel Analog Output Modules

750-550, -552, -554, -556, -560, -562, 563, -585, (and all variations),
753-550, -552, -554, -556

Table 377: 2 Channel Analog Output Modules

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	D1	D0	Output Value Channel 1
n+1	D3	D2	Output Value Channel 2

The output modules represent 2x2 bytes and seize 2 Instances in Class (0x68).

13.3.4.2 4 Channel Analog Output Modules

750-553, -555, -557, -559,
753-553, -555, -557, -559

Table 378: 4 Channel Analog Output Modules

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	D1	D0	Output Value Channel 1
n+1	D3	D2	Output Value Channel 2
n+2	D5	D4	Output Value Channel 3
n+3	D7	D6	Output Value Channel 4

The output modules represent 4x2 bytes and seize 4 Instances in Class (0x68).

13.3.5 Specialty Modules

WAGO has a host of Specialty I/O modules that perform various functions. With individual modules beside the data bytes also the control/status byte is mapped in the process image. The control/status byte is required for the bidirectional data exchange of the module with the higher-ranking control system. The control byte is transmitted from the control system to the module and the status byte from the module to the control system.

This allows, for example, setting of a counter with the control byte or displaying of overshooting or undershooting of the range with the status byte.

The control/status byte always lies in the low byte for the fieldbus coupler/controller with Ethernet/IP.

Information



Information to the structure of the Control/Status byte

For detailed information about the structure of a particular module's control/status byte, please refer to that module's manual. Manuals for each module can be found on the Internet under: <http://www.wago.com>.

The Specialty Modules represent as analog modules.

For this, the process input data of the Specialty Modules seize one Instance per channel in the Analog Input Point Object (Class 0x67) and the process output data seize one Instance seize one Instance in the Analog Input Point Object (Class 0x67) per channel in the Analog Output Point Object (Class 0x68).

13.3.5.1 Counter Modules

750-404, (and all variations except of /000-005),
753-404, (and variation /000-003)

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/status). The counter value is supplied as 32 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

Table 379: Counter Modules 750-404, (and all variations except of /000-005),
753-404, (and variation /000-003)

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	S	Status byte
	D1	D0	Counter value
	D3	D2	

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C	Control byte
	D1	D0	Counter setting value
	D3	D2	

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-404/000-005

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/ status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

Table 380: Counter Modules 750-404/000-005

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	S	Status byte
	D1	D0	Counter Value of Counter 1
	D3	D2	Counter Value of Counter 2

The specialty modules represent 2x3 bytes input data and seize 2 Instances in Class (0x67).

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C	Control byte
	D1	D0	Counter Setting Value of Counter 1
	D3	D2	Counter Setting Value of Counter 2

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-638,
753-638

The above Counter Modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 2 bytes of control/status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

Table 381: Counter Modules 750-638, 753-638

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	S0	Status byte von Counter 1
	D1	D0	Counter Value von Counter 1
n+1	-	S1	Status byte von Counter 2
	D3	D2	Counter Value von Counter 2

The specialty modules represent 2x3 bytes input data and seize 2 Instances in Class (0x67).

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C0	Control byte von Counter 1
	D1	D0	Counter Setting Value von Counter 1
n+1	-	C1	Control byte von Counter 2
	D3	D2	Counter Setting Value von Counter 2

And the specialty modules represent 2x3 bytes output data and seize 2 Instances in Class (0x68).

13.3.5.2 Pulse Width Modules

750-511, (and all variations /xxx-xxx)

The above Pulse Width modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of channel data and 2 bytes of control/status). The two channel values are supplied as 16 bits. Each channel has its own control/status byte. The following table illustrates the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

Table 382: Pulse Width Modules 750-511, /xxx-xxx

Input and Output Process			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C0/S0	Control/Status byte of Channel 1
	D1	D0	Data Value of Channel 1
n+1	-	C1/S1	Control/Status byte of Channel 2
	D3	D2	Data Value of Channel 2

The specialty modules represent 2x3 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

13.3.5.3 Serial Interface Modules with alternative Data Format

750-650, (and the variations /000-002, -004, -006, -009, -010, -011, -012, -013),
750-651, (and the variations /000-002, -003),
750-653, (and the variations /000-002, -007),
753-650, -653



Note

The process image of the / 003-000-variants depends on the parameterized operating mode!

With the freely parametrizable variations /003 000 of the serial interface modules, the desired operation mode can be set. Dependent on it, the process image of these modules is then the same, as from the appropriate variation.

The above Serial Interface Modules with alternative data format have a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 2 words mapped into each image. Word alignment is applied.

Table 383: Serial Interface Modules with alternative Data Format

Input and Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	D0	C/S	Data byte	Control/status byte
n+1	D2	D1	Data bytes	

The specialty modules represent 2x2 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

13.3.5.4 Serial Interface Modules with Standard Data Format

750-650/000-001, -014, -015, -016
750-651/000-001
750-653/000-001, -006

The above Serial Interface Modules with Standard Data Format have a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 3 words mapped into each image. Word alignment is applied.

Table 384: Serial Interface Modules with Standard Data Format

Input and Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	D0	C/S	Data byte	Control/status byte
	D2	D1	Data bytes	
	D4	D3		

The specialty modules represent 1x6 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.5.5 Data Exchange Module

750-654, (and the variation /000-001)

The Data Exchange modules have a total of 4 bytes of user data in both the Input and Output Process Image. The following tables illustrate the Input and Output Process Image, which has a total of 2 words mapped into each image. Word alignment is applied.

Table 385: Data Exchange Module

Input and Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	D1	D0	Data bytes
n+1	D3	D2	

The specialty modules represent 2x2 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

13.3.5.6 SSI Transmitter Interface Modules

750-630, (and all variations)



Note

The process image of the / 003-000-variants depends on the parameterized operating mode!

The operating mode of the configurable /003-000 I/O module versions can be set. Based on the operating mode, the process image of these I/O modules is then the same as that of the respective version.

The above SSI Transmitter Interface modules have a total of 4 bytes of user data in the Input Process Image, which has 2 words mapped into the image. Word alignment is applied.

Table 386: SSI Transmitter Interface Modules

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	D1	D0	Data bytes
n+1	D3	D2	

The specialty modules represent 2x2 bytes input data and seize 2 Instances in Class (0x67).

750-630/000-004, -005, -007

In the input process image, SSI transmitter interface modules with status occupy 5 usable bytes, 4 data bytes, and 1 additional status byte. A total of 3 words are assigned in the process image via word alignment.

Table 387: SSI Transmitter Interface I/O Modules with an Alternative Data Format

Input Process Image				
Instance	Byte Destination		Description	
	High Byte	High Byte		
n	-	S	not used	Status byte
	D1	D0	Data bytes	
	D3	D2		

The specialty modules represent 1x6 bytes and seize 1 Instance in Class (0x67).

13.3.5.7 Incremental Encoder Interface Modules

750-631/000-004, -010, -011

The above Incremental Encoder Interface modules have 5 bytes of input data and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which have 4 words into each image. Word alignment is applied.

Table 388: Incremental Encoder Interface Modules 750-631/000-004, -010, -011

Input Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	-	S	not used	Status byte
	D1	D0	Counter word	
	-	-	not used	
	D4	D3	Latch word	

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	-	C	Control byte of counter 1	
	D1	D0	Counter setting value of counter 1	
	-	-	not used	
	-	-	not used	

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-634

The above Incremental Encoder Interface module has 5 bytes of input data (6 bytes in cycle duration measurement mode) and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which has 4 words mapped into each image. Word alignment is applied.

Table 389: Incremental Encoder Interface Modules 750-634

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	S	not used Status byte
	D1	D0	Counter word
	-	(D2) *	not used (Periodic time)
	D4	D3	Latch word

*) If cycle duration measurement mode is enabled in the control byte, the cycle duration is given as a 24-bit value that is stored in D2 together with D3/D4.

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C	not used Control byte
	D1	D0	Counter setting word
	-	-	not used
	-	-	

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-637

The above Incremental Encoder Interface Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of encoder data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Table 390: Incremental Encoder Interface Modules 750-637

Input and Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C0/S0	Control/Status byte of Channel 1
	D1	D0	Data Value of Channel 1
n+1	-	C1/S1	Control/Status byte of Channel 2
	D3	D2	Data Value of Channel 2

The specialty modules represent 2x3 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

750-635,
753-635

The above Digital Pulse Interface module has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

Table 391: Incremental Encoder Interface Modules 750-635, 750-635

Input and Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	D0	C0/S0	Data byte	Control/status byte
	D2	D1	Data bytes	

The specialty modules represent 1x4 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.5.8 DC-Drive Controller

750-636

The DC-Drive Controller maps 6 bytes into both the input and output process image. The data sent and received are stored in up to 4 input and output bytes (D0 ... D3). Two control bytes (C0, C1) and two status bytes (S0/S1) are used to control the I/O module and the drive.

In addition to the position data in the input process image (D0 ... D3), it is possible to display extended status information (S2 ... S5). Then the three control bytes (C1 ... C3) and status bytes (S1 ... S3) are used to control the data flow.

Bit 3 of control byte C1 (C1.3) is used to switch between the process data and the extended status bytes in the input process image (Extended Info_ON). Bit 3 of status byte S1 (S1.3) is used to acknowledge the switching process.

Table 392: DC-Drive Controller 750-636

Input Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	S1	S0	Status byte S1	Status byte S0
	D1*) / S3**)	D0*) / S2**)	Actual position*) / Extended status byte S3**)	Actual position (LSB) / Extended status byte S2**)
	D3*) / S5**)	D2*) / S4**)	Actual position (MSB) / Extended status byte S3**)	Actual position*) / Extended status byte S4**)

*) ExtendedInfo_ON = '0'.

***) ExtendedInfo_ON = '1'.

Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	C1	C0	Control byte C1	Control byte C0
	D1	D0	Setpoint position	Setpoint position (LSB)
	D3	D2	Setpoint position (MSB)	Setpoint position

The specialty modules represent 1x6 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.5.9 Steppercontroller

750-670

The Steppercontroller RS422 / 24 V / 20 mA 750-670 provides the fieldbus coupler 12 bytes input and output process image via 1 logical channel. The data to be sent and received are stored in up to 7 output bytes (D0 ... D6) and 7 input bytes (D0 ... D6), depending on the operating mode.

Output byte D0 and input byte D0 are reserved and have no function assigned.

One I/O module control and status byte (C0, S0) and 3 application control and status bytes (C1 ... C3, S1 ... S3) provide the control of the data flow.

Switching between the two process images is conducted through bit 5 in the control byte (C0 (C0.5)). Activation of the mailbox is acknowledged by bit 5 of the status byte S0 (S0.5).

Table 393: Steppercontroller RS 422 / 24 V / 20 mA 750-670

Input Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	reserved	S0	reserved	Status byte S0
	D1	D0	Process data*) / Mailbox**)	
	D3	D2		
	D5	D4		
	S3	D6	Status byte S3	Process data*) / reserved**)
	S1	S2	Status byte S1	Status byte S2

*) Cyclic process image (Mailbox disabled)

***) Mailbox process image (Mailbox activated)

Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	reserved	C0	reserved	Control byte C0
	D1	D0	Process data*) / Mailbox**)	
	D3	D2		
	D5	D4		
	C3	D6	Control byte C3	Process data*) / reserved**)
	C1	C2	Control byte C1	Control byte C2

*) Cyclic process image (Mailbox disabled)

***) Mailbox process image (Mailbox activated)

The specialty modules represent 1x12 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.5.10 RTC Module

750-640

The RTC Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of module data and 1 byte of control/status and 1 byte ID for command). The following table illustrates the Input and Output Process Image, which have 3 words mapped into each image. Word alignment is applied.

Table 394: RTC Module 750-640

Input and Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	ID	C/S	Command byte	Control/status byte
	D1	D0	Data bytes	
	D3	D2		

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).and seize 1 Instance in Class (0x68).

13.3.5.11 DALI/DSI Master Module

750-641

The DALI/DSI Master module has a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of module data and 1 byte of control/status). The following tables illustrate the Input and Output Process Image, which have 3 words mapped into each image. Word alignment is applied.

Table 395: DALI/DSI Master module 750-641

Input Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	D0	S	DALI Response	Status byte
	D2	D1	Message 3	DALI Address
	D4	D3	Message 1	Message 2

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	D0	C	DALI command, DSI dimming value	Control byte
	D2	D1	Parameter 2	DALI Address
	D4	D3	Command extension	Parameter 1

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

13.3.5.12 EnOcean Radio Receiver

750-642

The EnOcean radio receiver has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following tables illustrate the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

Table 396: EnOcean Radio Receiver 750-642

Input Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	D0	S	Data byte	Status byte
n+1	D2	D1	Data bytes	

Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	-	C	not used	Control byte
n+1	-	-	not used	

The specialty modules represent 2x2 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

13.3.5.13 MP Bus Master Module

750-643

The MP Bus Master Module has a total of 8 bytes of user data in both the Input and Output Process Image (6 bytes of module data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Table 397: MP Bus Master Module 750-643

Input and Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	C1/S1	C0/S0	extended Control/Status byte	Control/status byte
	D1	D0	Data bytes	
	D3	D2		
	D5	D4		

The specialty modules represent 1x8 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.5.14 *Bluetooth*[®] RF-Transceiver

750-644

The size of the process image for the *Bluetooth*[®] module can be adjusted to 12, 24 or 48 bytes.

It consists of a control byte (input) or status byte (output); an empty byte; an overlayable mailbox with a size of 6, 12 or 18 bytes (mode 2); and the *Bluetooth*[®] process data with a size of 4 to 46 bytes.

Thus, each *Bluetooth*[®] module uses between 12 and 48 bytes in the process image. The sizes of the input and output process images are always the same.

The first byte contains the control/status byte; the second contains an empty byte.

Process data attach to this directly when the mailbox is hidden. When the mailbox is visible, the first 6, 12 or 18 bytes of process data are overlaid by the mailbox data, depending on their size. Bytes in the area behind the optionally visible mailbox contain basic process data. The internal structure of the *Bluetooth*[®] process data can be found in the documentation for the *Bluetooth*[®] 750-644 RF Transceiver.

The mailbox and the process image sizes are set with the startup tool WAGO-I/O-CHECK.

Table 398: Bluetooth® RF-Transceiver 750-644

Input and Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	-	C0/S0	not used	Control/status byte
	D1	D0	Mailbox (0, 3, 6 or 9 words) and Process data (2-23 words)	
	D3	D2		
	D5	D4		
		
D45	D44			

The 750-644 constitutes a special module, whose process data (12, 24 or 48 bytes) occupy on instances in classes 0x67 and 0x68.

13.3.5.15 Vibration Velocity/Bearing Condition Monitoring VIB I/O

750-645

The Vibration Velocity/Bearing Condition Monitoring VIB I/O has a total of 12 bytes of user data in both the Input and Output Process Image (8 bytes of module data and 4 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 8 words mapped into each image. Word alignment is applied.

Table 399: Vibration Velocity/Bearing Condition Monitoring VIB I/O 750-645

Input and Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	-	C0/S0	not used	Control/status byte (log. Channel 1, Sensor input 1)
	D1	D0	Data bytes (log. Channel 1, Sensor input 1)	
n+1	-	C1/S1	not used	Control/status byte (log. Channel 2, Sensor input 2)
	D3	D2	Data bytes (log. Channel 2, Sensor input 2)	
n+2	-	C2/S2	not used	Control/status byte (log. Channel 3, Sensor input 1)
	D5	D4	Data bytes (log. Channel 3, Sensor input 3)	
n+3	-	C3/S3	not used	Control/status byte (log. Channel 4, Sensor input 2)
	D7	D6	Data bytes (log. Channel 4, Sensor input 2)	

The specialty modules represent 4x3 bytes input and output data and seize 4 Instances in Class (0x67) and 4 Instances in Class (0x68).

13.3.5.16 AS-interface Master Module

750-655

The length of the process image of the AS-interface master module can be set to fixed sizes of 12, 20, 24, 32, 40 or 48 bytes.

It consists of a control or status byte, a mailbox with a size of 0, 6, 10, 12 or 18 bytes and the AS-interface process data, which can range from 0 to 32 bytes.

The AS-interface master module has a total of 6 to maximally 24 words data in both the Input and Output Process Image. Word alignment is applied.

The first Input and output word, which is assigned to an AS-interface master module, contains the status / control byte and one empty byte.

Subsequently the mailbox data are mapped, when the mailbox is permanently superimposed (Mode 1).

In the operating mode with suppressible mailbox (Mode 2), the mailbox and the cyclical process data are mapped next.

The following words contain the remaining process data.

The mailbox and the process image sizes are set with the startup tool WAGO-I/O-CHECK.

Table 400: AS-interface Master module 750-655

Input and Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C0/S0	not used Control/status byte
	D1	D0	Mailbox (0, 3, 5, 6 or 9 words)/ Process data (0-16 words)
	D3	D2	
	D5	D4	
	
	D45	D44	

The specialty modules represent 1x 12...48 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.6 System Modules

13.3.6.1 System Modules with Diagnostics

750-610, -611

The modules provide 2 bits of diagnostics in the Input Process Image for monitoring of the internal power supply.

Table 401: System Modules with Diagnostics 750-610, -611

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 2 Fuse	Diagnostic bit S 1 Fuse

The system modules seize 2 Instances in Class (0x65).

13.3.6.2 Binary Space Module

750-622

The Binary Space Modules behave alternatively like 2 channel digital input modules or output modules and seize depending upon the selected settings 1, 2, 3 or 4 bits per channel. According to this, 2, 4, 6 or 8 bits are occupied then either in the process input or the process output image.

Table 402: Binary Space Module 750-622 (with behavior like 2 channel digital input)

Input and Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(Data bit DI 8)	(Data bit DI 7)	(Data bit DI 6)	(Data bit DI 5)	(Data bit DI 4)	(Data bit DI 3)	Data bit DI 2	Data bit DI 1

The Binary Space Modules seize 2, 4, 6 or 8 Instances in class (0x65) or in Class (0x66).

14 Application Examples

14.1 Test of MODBUS protocol and fieldbus nodes

You require a MODBUS master to test the function of your fieldbus node. For this purpose, various manufacturers offer a range of PC applications that you can, in part, download from the Internet as free of charge demo versions.

One of the programs which is particularly suitable to test your ETHERNET TCP/IP fieldbus node, is for instance **ModScan** from Win-Tech.

Information



Additional Information

A free of charge demo version from ModScan32 and further utilities from Win-Tech can be found in the Internet under:

<http://www.win-tech.com/html/demos.htm>

ModScan32 is a Windows application that works as a MODBUS master.

This program allows you to access the data points of your connected ETHERNET TCP/IP fieldbus node and to proceed with the desired changes.

Information



Additional Information

For a description example relating to the software operation, refer to:

<http://www.win-tech.com/html/modscan32.htm>

14.2 Visualization and Control using SCADA Software

This chapter is intended to give insight into how the WAGO ETHERNET fieldbus coupler/controller can be used for process visualization and control using standard user software.

There is a wide range of process visualization programs, called SCADA Software, from various manufacturers.

Information



Additional Information

For a selection of SCADA products, look under i.e.:

<http://www.abpubs.demon.co.uk/scadasites.htm>

SCADA is the abbreviation for Supervisory Control and Data Acquisition.

It is a user-orientated tool used as a production information system in the areas of automation technology, process control and production monitoring.

The use of SCADA systems includes the areas of visualization and monitoring, data access, trend recording, event and alarm processing, process analysis and targeted intervention in a process (control).

The WAGO ETHERNET fieldbus node provides the required process input and output values.

Note



SCADA software has to provide a MODBUS device driver and support MODBUS/TCP functions!

When choosing suitable SCADA software, ensure that it provides a MODBUS device driver and supports the MODBUS/TCP functions in the coupler.

Visualization programs with MODBUS device drivers are available from i.e. Wonderware, National Instruments, Think&Do or KEPware Inc., some of which are available on the Internet as demo versions.

The operation of these programs is very specific. However, a few essential steps are described to illustrate the way an application can be developed using a WAGO ETHERNET fieldbus node and SCADA software in principle:

1. Load the MODBUS ETHERNET driver and select MODBUS ETHERNET
2. Enter the IP address for addressing the fieldbus node

At this point, some programs allow the user to give the node an alias name, i.e. to call the node "Measuring data". The node can then be addressed with this name.

3. Create a graphic object, such as a switch (digital) or a potentiometer (analog)

This object is displayed on the work area.

4. Link the object to the desired data point on the node by entering the following data:
 - Node address (IP address or alias name)
 - The desired MODBUS function codes (register/bit read/write)
 - The MODBUS address of the selected channel

Entry is program specific.

Depending on the user software the MODBUS addressing of a bus module can be represented with up to 5 digits.

Example of the MODBUS Addressing

In the case of SCADA Software Lookout from National Instruments the MODBUS function codes are used with a 6 digit coding, whereby the first digit represents the MODBUS table (0, 1, 3 or 4) and implicit the function code (see following table):

Table 403: MODBUS table and function codes

MODBUS table	MODBUS function code	
0	FC1 or FC15	Reading of input bits or writing of several output bits
1	FC2	Reading of several input bits
3	FC4 or FC 16	Reading of several input registers or writing of several output registers
4	FC3	Reading of several input registers

The following five digits specify the channel number (beginning with 1) of the consecutively numbered digital or analog input and/or output channels.

Examples:

- Reading/writing the first digital input: i.e. 0 0000 1
- Reading/writing the second analog input: i.e. 3 0000 2

Application Example:

Thus, the digital input channel 2 of the above node "Measuring data" can be read out with the input: "Measuring data. 0 0000 2".

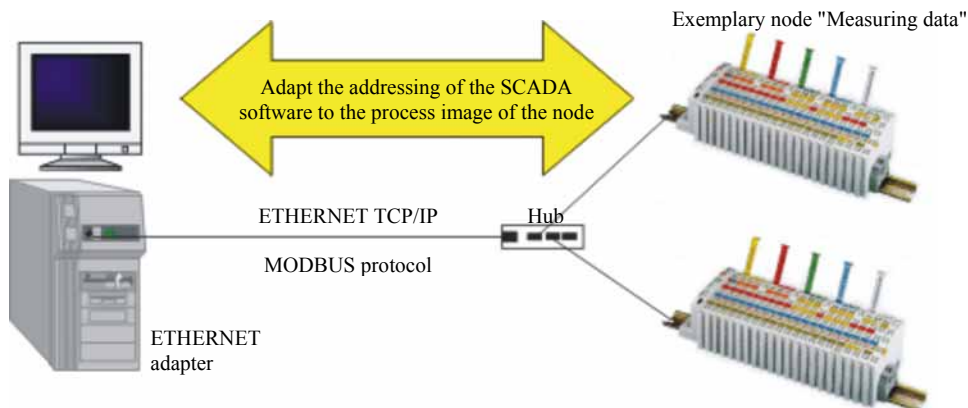


Figure 77: Example SCADA software with MODBUS driver

Information



Additional Information

Please refer to the respective SCADA product manual for a detailed description of the particular software operation.

15 Use in Hazardous Environments

The **WAGO-I/O-SYSTEM 750** (electrical equipment) is designed for use in Zone 2 hazardous areas.

The following sections include both the general identification of components (devices) and the installation regulations to be observed. The individual subsections of the "Installation Regulations" section must be taken into account if the I/O module has the required approval or is subject to the range of application of the ATEX directive.

15.1 Marking Configuration Examples

15.1.1 Marking for Europe according to CENELEC and IEC

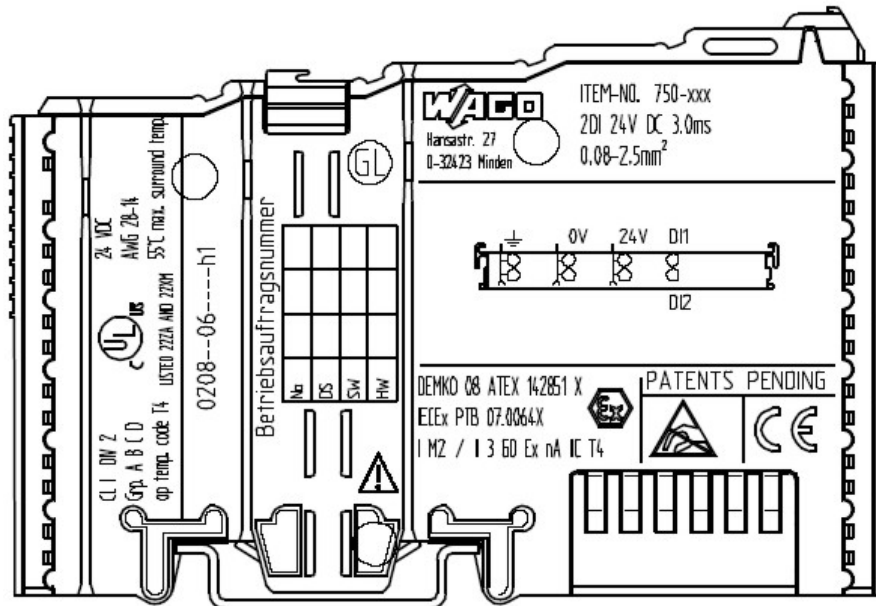


Figure 78: Side marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

DEMKO 08 ATEX 142851 X
IECEX PTB 07.0064X
I M2 / II 3 GD Ex nA IIC T4



Figure 79: Printing Text detail – Marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

Table 404: Description of marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

Printing on Text	Description
DEMKO 08 ATEX 142851 X IECEX PTB 07.0064X	Approval body and/or number of the examination certificate
I M2 / II 3 GD	Explosion protection group and Unit category
Ex nA	Type of ignition and extended identification
IIC	Explosion protection group
T4	Temperature class

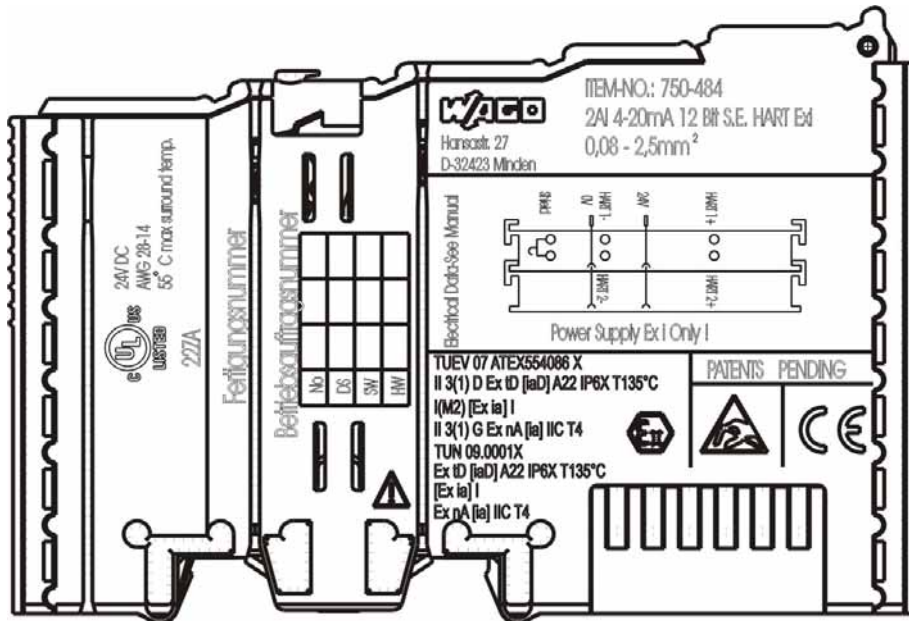


Figure 80: Side marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

TUEV 07 ATEX554086 X
II 3(1) D Ex tD [iaD] A22 IP6X T135°C
I(M2) [Ex ia] I
II 3(1) G Ex nA [ia] IIC T4
TUN 09.0001X
Ex tD [iaD] A22 IP6X T135°C
[Ex ia] I
Ex nA [ia] IIC T4



Figure 81: Text detail – Marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

Table 405: Description of marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

Inscription text	Description
TÜV 07 ATEX 554086 X TUN 09.0001X	Approving authority or certificate numbers
Dust	
II	Device group: All except mining
3(1)D	Device category: Zone 22 device (Zone 20 subunit)
Ex	Explosion protection mark
tD	Protection by enclosure
[iaD]	Approved in accordance with "Dust intrinsic safety" standard
A22	Surface temperature determined according to Procedure A, use in Zone 22
IP6X	Dust-tight (totally protected against dust)
T 135°C	Max. surface temp. of the enclosure (no dust bin)
Mining	
I	Device group: Mining
(M2)	Device category: High degree of safety
[Ex ia]	Explosion protection: Mark with category of type of protection intrinsic safety: Even safe when two errors occur
I	Device group: Mining
Gases	
II	Device group: All except mining
3(1)G	Device category: Zone 2 device (Zone 0 subunit)
Ex	Explosion protection mark
nA	Type of protection: Non-sparking operating equipment
[ia]	Category of type of protection intrinsic safety: Even safe when two errors occur
IIC	Explosion Group
T4	Temperature class: Max. surface temperature 135°C

15.1.2 Marking for America according to NEC 500

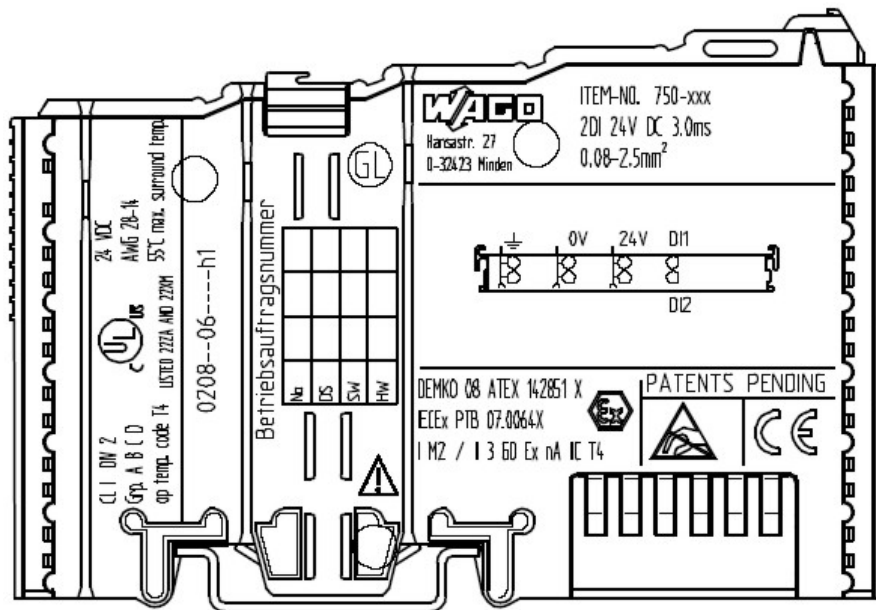


Figure 82: Side marking example for I/O modules according to NEC 500



Figure 83: Text detail – Marking example for I/O modules according to NEC 500

Table 406: Description of marking example for I/O modules according to NEC 500

Printing on Text	Description
CL 1	Explosion protection group (condition of use category)
DIV 2	Area of application (zone)
Grp. ABCD	Explosion group (gas group)
Optemp code T4	Temperature class

15.2 Installation Regulations

In the **Federal Republic of Germany**, various national regulations for the installation in explosive areas must be taken into consideration. The basis for this forms the working reliability regulation, which is the national conversion of the European guideline 99/92/E6. They are complemented by the installation regulation EN 60079-14. The following are excerpts from additional VDE regulations:

Table 407: VDE Installation Regulations in Germany

DIN VDE 0100	Installation in power plants with rated voltages up to 1000 V
DIN VDE 0101	Installation in power plants with rated voltages above 1 kV
DIN VDE 0800	Installation and operation in telecommunication plants including information processing equipment
DIN VDE 0185	lightning protection systems

The **USA** and **Canada** have their own regulations. The following are excerpts from these regulations:

Table 408: Installation Regulations in USA and Canada

NFPA 70	National Electrical Code Art. 500 Hazardous Locations
ANSI/ISA-RP 12.6-1987	Recommended Practice
C22.1	Canadian Electrical Code

NOTICE

Notice the following points

When using the **WAGO-I/O SYSTEM 750** (electrical operation) with Ex approval, the following points are mandatory:

15.2.1 Special Conditions for Safe Operation of the ATEX and IEC Ex (acc. DEMKO 08 ATEX 142851X and IECEx PTB 07.0064)

The fieldbus-independent I/O modules of the WAGO-I/O-SYSTEM 750-.../...-... must be installed in an environment with degree of pollution 2 or better. In the final application, the I/O modules must be mounted in an enclosure with IP 54 degree of protection at a minimum with the following exceptions:

- I/O modules 750-440, 750-609 and 750-611 must be installed in an IP 64 minimum enclosure.
- I/O module 750-540 must be installed in an IP 64 minimum enclosure for 230 V AC applications.
- I/O module 750-440 may be used up to max. 120 V AC.

When used in the presence of combustible dust, all devices and the enclosure shall be fully tested and assessed in compliance with the requirements of IEC 61241-0:2004 and IEC 61241-1:2004.

When used in mining applications the equipment shall be installed in a suitable enclosure according to EN 60079-0:2006 and EN 60079-1:2007.

I/O modules fieldbus plugs or fuses may only be installed, added, removed or replaced when the system and field supply is switched off or the area exhibits no explosive atmosphere.

DIP switches, coding switches and potentiometers that are connected to the I/O module may only be operated if an explosive atmosphere can be ruled out.

I/O module 750-642 may only be used in conjunction with antenna 758-910 with a max. cable length of 2.5 m.

To exceed the rated voltage no more than 40%, the supply connections must have transient protection.

The permissible ambient temperature range is 0 °C to +55 °C.

15.2.2 Special conditions for safe use (ATEX Certificate TÜV 07 ATEX 554086 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the field bus independent I/O modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15, EN 61241-0 and EN 61241-1. For use as group I, electrical apparatus M2, the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.
2. If the interface circuits are operated without the field bus coupler station type 750-3./...-... (DEMKO 08 ATEX 142851 X), measures must be taken outside of the device so that the rating voltage is not being exceeded of more than 40% because of transient disturbances.
3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “CF-Card”, “USB”, “Fieldbus connection“, “Configuration and programming interface“, “antenna socket“, “D-Sub“ and the “Ethernet interface“. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.
6. For the type 750-601 the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The ambient temperature range is: $0^{\circ}\text{C} \leq T_a \leq +55^{\circ}\text{C}$ (for extended details please note certificate).

8. The following warnings shall be placed nearby the unit:

 **WARNING**

Do not remove or replace fuse when energized!

If the module is energized do not remove or replace the fuse.

 **WARNING**

Do not separate when energized!

Do not separate the module when energized!

 **WARNING**

Separate only in a non-hazardous area!

Separate the module only in a non-hazardous area!

15.2.3 Special conditions for safe use (IEC-Ex Certificate TUN 09.0001 X)

1. For use as Dc- or Gc-apparatus (in zone 2 or 22) the fieldbus independent I/O modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15, IEC 61241-0 and IEC 61241-1. For use as group I, electrical apparatus M2, the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40% because of transient disturbances.
3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “CF-Card”, “USB”, “Fieldbus connection“, “Configuration and programming interface“, “antenna socket“, “D-Sub“ and the “Ethernet interface“. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.
6. For the type 750-601 the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The ambient temperature range is: $0^{\circ}\text{C} \leq T_a \leq +55^{\circ}\text{C}$ (For extensions please see the certificate).

8. The following warnings shall be placed nearby the unit:

 **WARNING**

Do not remove or replace fuse when energized!

If the module is energized do not remove or replace the fuse.

 **WARNING**

Do not separate when energized!

Do not separate the module when energized!

 **WARNING**

Separate only in a non-hazardous area!

Separate the module only in a non-hazardous area!

15.2.4 ANSI/ISA 12.12.01

This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only.

This equipment is to be fitted within tool-secured enclosures only.

WARNING

Explosion hazard!

Explosion hazard - substitution of components may impair suitability for Class I, Div. 2.

WARNING

Disconnect device when power is off and only in a non-hazardous area!

Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous near each operator accessible connector and fuse holder." When a fuse is provided, the following information shall be provided: "A switch suitable for the location where the equipment is installed shall be provided to remove the power from the fuse."

For devices with Ethernet connectors:

"Only for use in LAN, not for connection to telecommunication circuits".

WARNING

Use only with antenna module 758-910!

Use Module 750-642 only with antenna module 758-910.

For Couplers/Controllers and Economy bus modules only: "The configuration Interface Service connector is for temporary connection only. Do not connect or disconnect unless the area is known to be nonhazardous. Connection or disconnection in an explosive atmosphere could result in an explosion.

WARNING

Devices containing fuses must not be fitted into circuits subject to over loads!

Devices containing fuses must not be fitted into circuits subject to over loads, e.g. motor circuits!

 **WARNING**

Do not connect or disconnect SD-Card unless the area known to be free of ignitable concentrations of flammable gases or vapors!

Do not connect or disconnect SD-Card while circuit is live unless the area is known to be free of ignitable concentrations of flammable gases or vapors.

 **Information**

Additional Information

Proof of certification is available on request. Also take note of the information given on the module technical information sheet. The Instruction Manual, containing these special conditions for safe use, must be readily available to the user.

16 Appendix

16.1 MIB II Groups

16.1.1 System Group

The system group contains general information about the coupler/controller.

Table 409: MIB II – System group

Identifier	Entry	Access	Description
1.3.6.1.2.1.1.1	sysDescr	R	This entry contains the device identification. The object has a fixed code (e.g., "WAGO 750-841")
1.3.6.1.2.1.1.2	sysObjectID	R	This entry contains the manufacturer's authorization identification.
1.3.6.1.2.1.1.3	sysUpTime	R	This entry contains the time (in hundredths of a second) since the management unit has been last reset.
1.3.6.1.2.1.1.4	sysContakt	R/W	This entry contains the identification and contact information for the system contact person.
1.3.6.1.2.1.1.5	sysName	R/W	This entry contains the administration-assigned device name.
1.3.6.1.2.1.1.6	sysLocation	R/W	This entry contains the node's physical location.
1.3.6.1.2.1.1.7	sysServices	R	This entry designates the quantity of services that this coupler/controller contains.

16.1.2 Interface Group

The interface group contains information and statistics about the device interface.

A device interface describes the Ethernet interface of a coupler/controller and provides status information on the physical Ethernet ports as well as on the internal loopback interface.

Table 410: MIB II – Interface Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.2.1	ifNumber	R	Number of network interfaces in this system
1.3.6.1.2.1.2.2	ifTable	-	List of network interfaces
1.3.6.1.2.1.2.2.1	ifEntry	-	Network interface entry
1.3.6.1.2.1.2.2.1.1	ifIndex	R	This entry contains a unique value for each interface
1.3.6.1.2.1.2.2.1.2	ifDescr	R	This entry contains the name of the manufacturer, the product name, and the version of the hardware interface: e.g., "WAGO Kontakttechnik GmbH 750-841: Rev 1.0"
1.3.6.1.2.1.2.2.1.3	ifType	R	This entry describes the type of interface. ETHERNET CSMA/CD = 6 Software Loopback = 24
1.3.6.1.2.1.2.2.1.4	ifMtu	R	This entry specifies the largest transfer unit; i.e., the maximum telegram length that can be transferred via this interface.
1.3.6.1.2.1.2.2.1.5	ifSpeed	R	This entry indicates the interface speed in bits per second.
1.3.6.1.2.1.2.2.1.6	ifPhysAddress	R	This entry indicates the physical address of the interface. For example, for Ethernet, this entry contains a MAC ID.
1.3.6.1.2.1.2.2.1.7	ifAdmin-Status	R/W	This entry specifies the desired state of the interfaces. Possible values are: up(1): Ready for operation for transmission and reception down(2): Interface is switched off testing(3): Interface is in test mode
1.3.6.1.2.1.2.2.1.8.	ifOperStatus	R	This entry indicates the current operational state of the interface.
1.3.6.1.2.1.2.2.1.9.	ifLastChange	R	This entry indicates the value of the sysUpTime when the state was last changed.
1.3.6.1.2.1.2.2.1.10	ifInOctets	R	This entry gives the total number of bytes received via interface.
1.3.6.1.2.1.2.2.1.11	ifInUcastPkts	R	This entry indicates the number of received unicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.12	ifInNUcastPkts	R	This entry indicates the number of received broad and multicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.13	ifInDiscards	R	This entry indicates the number of packets that were discarded even though no errors had been detected.
1.3.6.1.2.1.2.2.1.14	ifInErrors	R	This entry indicates the number of received packets that contained errors preventing them from being deliverable to a higher layer.

Table 410: MIB II – Interface Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.2.2.1.15	IfInUnknown-Protos	R	This entry indicates the number of received packets sent to an unknown or unsupported port number.
1.3.6.1.2.1.2.2.1.16	ifOutOctets	R	This entry gives the total number of bytes sent via interface.
1.3.6.1.2.1.2.2.1.17	ifOutUcastPkts	R	This entry contains the number of outgoing unicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.18	ifOutNUcastPkts	R	This entry indicates the number of outgoing broad and multicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.19	ifOutDiscards	R	This entry indicates the number of packets that were discarded even though no errors had been detected.
1.3.6.1.2.1.2.2.1.20	ifOutErrors	R	This entry indicates the number of packets that could not be transmitted because of errors.

16.1.3 IP Group

The IP group contains information about IP communication.

Table 411: MIB II – IP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.4.1	ipForwarding	R/W	1: Host is a router; 2: Host is not a router
1.3.6.1.2.1.4.2	ipDefaultTTL	R/W	Default value for the Time-To-Live field of each IP frame.
1.3.6.1.2.1.4.3	ipInReceives	R	Number of received IP frames, including those received in error.
1.3.6.1.2.1.4.4	ipInHdrErrors	R	Number of received IP frames with header errors.
1.3.6.1.2.1.4.5	ipInAddrErrors	R	Number of received IP frames with a misdirected IP address.
1.3.6.1.2.1.4.6	ipForwDatagrams	R	Number of received IP frames passed on (routed)
1.3.6.1.2.1.4.7	ipUnknownProtos	R	Number of received IP frames with an unknown protocol type.
1.3.6.1.2.1.4.8	ipInDiscards	R	Number of received IP frames rejected although no disturbance was present.
1.3.6.1.2.1.4.9	ipInDelivers	R	Number of received IP frames passed on a higher protocol layer.
1.3.6.1.2.1.4.10	ipOutRequests	R	Number of sent IP frames
1.3.6.1.2.1.4.11	ipOutDiscards	R	Number of rejected IP Frames that should have been sent.
1.3.6.1.2.1.4.12	ipOutNoRoutes	R	Number of sent IP frames rejected because of incorrect routing information.
1.3.6.1.2.1.4.13	ipReasmTimeout	R	Minimum time duration until an IP frame is re-assembled.
1.3.6.1.2.1.4.14	ipReasmReqds	R	Minimum number of the IP fragments for building up and passing on.
1.3.6.1.2.1.4.15	ipReasmOKs	R	Number of IP frames re-assembled successfully.
1.3.6.1.2.1.4.16	ipReasmFails	R	Number of IP frames not re-assembled successfully.
1.3.6.1.2.1.4.17	ipFragOKs	R	Number of IP frames fragmented and passed on.
1.3.6.1.2.1.4.18	ipFragFails	R	Number of IP frames that should have been fragmented but could not be, because their don't fragment bit was set in the header.
1.3.6.1.2.1.4.19	ipFragCreates	R	Number of generated IP fragment frames
1.3.6.1.2.1.4.20	ipAddrTable	-	Table of all local IP addresses of the coupler/controller.
1.3.6.1.2.1.4.20.1	ipAddrEntry	-	Address information for an entry
1.3.6.1.2.1.4.20.1.1	ipAdEntAddr	R	The IP address corresponding to the entry's address information
1.3.6.1.2.1.4.20.1.2	ipAdEntIfIndex	R	Index of the interface
1.3.6.1.2.1.4.20.1.3	ipAdEntNetMask	R	The entry's associated subnet mask
1.3.6.1.2.1.4.20.1.4	ipAdEntBcastAddr	R	Value of the last significant bit in the IP broadcast address
1.3.6.1.2.1.4.20.1.5	IpAdEntReasm- MaxSize	R	The size of the longest IP telegram that can be defragmented (reassembled) again.
1.3.6.1.2.1.4.23	ipRoutingDiscards	R	Number of deleted routing entries

16.1.4 IpRoute Table Group

The IP route table contains information about the routing table in the coupler/controller.

Table 412: MIB II – IpRoute Table Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.4.21	ipRouteTable	-	IP routing table
1.3.6.1.2.1.4.21.1	ipRouteEntry	-	A routing entry for a particular destination
1.3.6.1.2.1.4.21.1.1	ipRouteDest	R/W	This entry indicates the destination address of the routing entry
1.3.6.1.2.1.4.21.1.2	ipRouteIfIndex	R/W	This entry indicates the index of the interface, which is the next route destination
1.3.6.1.2.1.4.21.1.3	ipRouteMetric1	R/W	The primary route to the target system
1.3.6.1.2.1.4.21.1.4	ipRouteMetric2	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.5	ipRouteMetric3	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.6	ipRouteMetric4	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.7	ipRouteNextHop	R/W	The IP address of the next route section
1.3.6.1.2.1.4.21.1.8	ipRouteType	R/W	The route type
1.3.6.1.2.1.4.21.1.9	ipRouteProto	R	Routing mechanism via which the route is developed
1.3.6.1.2.1.4.21.1.10	ipRouteAge	R/W	Number of seconds since then the route was last renewed/examined
1.3.6.1.2.1.4.21.1.11	ipRouteMask	R/W	This entry contents the subnet mask for this entry
1.3.6.1.2.1.4.21.1.12	ipRouteMetric5	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.13	ipRouteInfo	R/W	A reference to a special MIB

16.1.5 ICMP Group

Table 413: MIB II – ICMP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.5.1	icmpInMsgs	R	Number of received ICMP messages
1.3.6.1.2.1.5.2	icmpInErrors	R	Number of received ICMP errors containing ICMP-specific errors
1.3.6.1.2.1.5.3	icmpInDestUnreachs	R	Number of received ICMP destination unreachable messages
1.3.6.1.2.1.5.4	icmpInTimeExcds	R	Number of received ICMP time exceeded messages
1.3.6.1.2.1.5.5	icmpInParmProbs	R	Number of received ICMP parameter problem messages
1.3.6.1.2.1.5.6	icmpInSrcQuenchs	R	Number of received ICMP source quench messages
1.3.6.1.2.1.5.7	icmpInRedirects	R	Number of received ICMP redirect messages
1.3.6.1.2.1.5.8	icmpInEchos	R	Number of received ICMP echo request messages (Ping)
1.3.6.1.2.1.5.9	icmpInEchoReps	R	Number of received ICMP echo reply messages (Ping)
1.3.6.1.2.1.5.10	icmpInTimestamps	R	Number of received ICMP timestamp request messages
1.3.6.1.2.1.5.11	icmpInTimestampReps	R	Number of received ICMP timestamp reply messages
1.3.6.1.2.1.5.12	icmpInAddrMasks	R	Number of received ICMP address mask request messages
1.3.6.1.2.1.5.13	icmpInAddrMaskReps	R	Number of received ICMP address mask reply messages
1.3.6.1.2.1.5.14	icmpOutMsgs	R	Number of sent ICMP messages
1.3.6.1.2.1.5.15	icmpOutErrors	R	Number of sent ICMP messages that could not be sent due to errors
1.3.6.1.2.1.5.16	icmpOutDestUnreachs	R	Number of sent ICMP destination unreachable messages
1.3.6.1.2.1.5.17	icmpOutTimeExcds	R	Number of sent ICMP time exceeded messages
1.3.6.1.2.1.5.18	icmpOutParmProbs	R	Number of sent ICMP parameter problem messages
1.3.6.1.2.1.5.19	icmpOutSrcQuenchs	R	Number of sent ICMP source quench messages
1.3.6.1.2.1.5.20	icmpOutRedirects	R	Number of sent ICMP redirection messages
1.3.6.1.2.1.5.21	icmpOutEchos	R	Number of sent ICMP echo request messages
1.3.6.1.2.1.5.22	icmpOutEchoReps	R	Number of sent ICMP echo reply messages
1.3.6.1.2.1.5.23	icmpOutTimestamps	R	Number of sent ICMP timestamp request messages
1.3.6.1.2.1.5.24	icmpOutTimestampReps	R	Number of sent ICMP timestamp reply messages
1.3.6.1.2.1.5.25	icmpOutAddrMasks	R	Number of sent ICMP address mask request messages
1.3.6.1.2.1.5.26	icmpOutAddrMaskReps	R	Number of sent ICMP address mask reply messages

16.1.6 TCP Group

Table 414: MIB II – TCP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.6.1	tcpRtoAlgorithm	R	Retransmission time: 1 = other, 2 = constant, 3 = RSRE, 4 = VANJ
1.3.6.1.2.1.6.2	tcpRtoMin	R	Minimum value for the retransmission timer
1.3.6.1.2.1.6.3	tcpRtoMax	R	Maximum value for the retransmission timer
1.3.6.1.2.1.6.4	tcpMaxConn	R	Number of maximum TCP connections that can exist simultaneously
1.3.6.1.2.1.6.5	tcpActiveOpens	R	Number of existing active TCP connections
1.3.6.1.2.1.6.6	tcpPassiveOpens	R	Number of existing passive TCP connections
1.3.6.1.2.1.6.7	tcpAttemptFails	R	Number of failed connection attempts
1.3.6.1.2.1.6.8	tcpEstabResets	R	Number of connection resets
1.3.6.1.2.1.6.9	tcpCurrEstab	R	The number of TCP connections for which the current state is either Established or Close-Wait
1.3.6.1.2.1.6.10	tcpInSegs	R	Number of received TCP frames including the error frames
1.3.6.1.2.1.6.11	tcpOutSegs	R	Number of correctly sent TCP frames with data
1.3.6.1.2.1.6.12	tcpRetransSegs	R	Number of sent TCP frames retransmitted because of errors
1.3.6.1.2.1.6.13	tcpConnTable	-	For each existing connection, a table entry is created
1.3.6.1.2.1.6.13.1	tcpConnEntry	-	Table entry for connection
1.3.6.1.2.1.6.13.1.1	tcpConnState	R	This entry indicates the status of the TCP connection
1.3.6.1.2.1.6.13.1.2	tcpConnLocalAddress	R	The entry contains the IP address for the connection. For a server, this entry is constant 0.0.0.0
1.3.6.1.2.1.6.13.1.3	tcpConnLocalPort	R	The entry indicates the port number of the TCP connection.
1.3.6.1.2.1.6.13.1.4	tcpConnRemAddress	R	The entry contains the remote IP address of the TCP connection.
1.3.6.1.2.1.6.13.1.5	tcpConnRemPort	R	The entry contains the remote port of the TCP connection.
1.3.6.1.2.1.6.14	tcpInErrs	R	Number of received incorrect TCP frames
1.3.6.1.2.1.6.15	tcpOutRsts	R	Number of sent TCP frames with set RST flag

16.1.7 UDP Group

Table 415: MIB II – UDP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.7.1	udpInDatagrams	R	Number of received UDP frames that could be passed on to the appropriate applications
1.3.6.1.2.1.7.2	udpNoPorts	R	Number of received UDP frames that could not be passed on to the appropriate applications (port unreachable)
1.3.6.1.2.1.7.3	udpInErrors	R	Number of received UDP frames that could not be passed on to the appropriate applications for other reasons.
1.3.6.1.2.1.7.4	udpOutDatagrams	R	Number of sent UDP frames
1.3.6.1.2.1.7.5	udpTable	-	A table entry is created for each application that received UDP frames
1.3.6.1.2.1.7.5.1	udpEntry	-	Table entry for an application that received an UDP frame
1.3.6.1.2.1.7.5.1.1	udpLocalAddress	R	IP address of the local UDP server
1.3.6.1.2.1.7.5.1.2	udpLocalPort	R	Port number of the local UDP server

16.1.8 SNMP Group

Table 416: MIB II – SNMP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.11.1	snmpInPkts	R	Number of received SNMP frames
1.3.6.1.2.1.11.2	snmpOutPkts	R	Number of sent SNMP frames
1.3.6.1.2.1.11.3	snmpInBadVersions	R	Number of received SNMP frames with an invalid version number
1.3.6.1.2.1.11.4	snmpInBadCommunity-Names	R	Number of received SNMP frames with an invalid community
1.3.6.1.2.1.11.5	snmpInBadCommunity Uses	R	Number of received SNMP frames whose community did not have sufficient authorization for the actions that it tried to execute
1.3.6.1.2.1.11.6	snmpInASNParseErrs	R	Number of received SNMP frames with an incorrect structure
1.3.6.1.2.1.11.8	snmpInTooBigs	R	Number of received SNMP frames that acknowledged the result too Big
1.3.6.1.2.1.11.9	snmpInNoSuchNames	R	Number of received SNMP frames that acknowledged the result noSuchName
1.3.6.1.2.1.11.10	snmpInBadValues	R	Number of received SNMP frames that acknowledged the result bad value
1.3.6.1.2.1.11.11	snmpInReadOnlys	R	Number of received SNMP frames that acknowledged the result readOnly
1.3.6.1.2.1.11.12	snmpInGenErrs	R	Number of received SNMP frames that acknowledged the result genError
1.3.6.1.2.1.11.13	snmpInTotalReqVars	R	Number of received SNMP frames with valid GET or GET-NEXT requests
1.3.6.1.2.1.11.14	snmpInTotalSetVars	R	Number of received SNMP frames with valid SET requests
1.3.6.1.2.1.11.15	snmpInGetRequests	R	Number of GET requests received and processed
1.3.6.1.2.1.11.16	snmpInGetNexts	R	Number of GET-NEXT requests received and processed
1.3.6.1.2.1.11.17	snmpInSetRequests	R	Number of SET requests received and processed
1.3.6.1.2.1.11.18	snmpInGetResponses	R	Number of received GET responses
1.3.6.1.2.1.11.19	snmpInTraps	R	Number of received traps
1.3.6.1.2.1.11.20	snmpOutTooBigs	R	Number of sent SNMP frames that contained the result too Big
1.3.6.1.2.1.11.21	snmpOutNoSuchNames	R	Number of sent SNMP frames that contained the result noSuchName
1.3.6.1.2.1.11.22	snmpOutBadValues	R	Number of sent SNMP frames that contained the result bad value
1.3.6.1.2.1.11.24	SnmpOutGenErrs	R	Number of sent SNMP frames that contained the result genErrs
1.3.6.1.2.1.11.25	snmpOutGetRequests	R	Number of GET requests sent
1.3.6.1.2.1.11.26	SnmpOutGetNexts	R	Number of GET NEXT requests sent
1.3.6.1.2.1.11.27	snmpOutSetRequests	R	Number of SET requests sent
1.3.6.1.2.1.11.28	snmpOutGetResponses	R	Number of GET responses sent
1.3.6.1.2.1.11.29	snmpOutTraps	R	Number of traps sent
1.3.6.1.2.1.11.30	snmpEnableAuthenTraps	R/W	Authentication failure traps(1 = on, 2 = off)

16.2 WAGO MIB Groups

16.2.1 Company Group

The company group contains general information about the company WAGO Kontakttechnik GmbH & Co. KG.

Table 417: WAGO MIB – Company Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.1.1	wagoName	R	Company's registered name Default value: "WAGO Kontakttechnik GmbH & Co. KG"
1.3.6.1.4.1.13576.1.2	wagoDescription	R	Description of company Default value: "WAGO Kontakttechnik GmbH & Co. KG, Hansastr. 27, D-32423 Minden"
1.3.6.1.4.1.13576.1.3	wagoURL	R	URL for company web site Default value: "www.wago.com"

16.2.2 Product Group

The product group contains information about the controller.

Table 418: WAGO MIB – Product Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.1	wioArticleName	R	Name of article Default value: "750-8xx/000-000"
1.3.6.1.4.1.13576.10.1.2	wioArticleDescription	R	Description of article Default value: "WAGO Ethernet (10/100MBit) FBC"
1.3.6.1.4.1.13576.10.1.3	wioSerialNumber	R	Serial number of article Default value: "SNxxxxxxx-Txxxxxx-mac 0030DExxxxxx"
1.3.6.1.4.1.13576.10.1.4	wioMacAddress	R	MAC address of article Default value: "0030DExxxxxx"
1.3.6.1.4.1.13576.10.1.5	wioURLDatasheet	R	URL to datasheet of article Default value: "http://www.wago.com/wagoweb/documentation/navigate/nm0dc_e.htm#ethernet"
1.3.6.1.4.1.13576.10.1.6	wioURLManual	R	URL to manual of article Default value: "http://www.wago.com/wagoweb/documentation/navigate/nm0dc_e.htm#ethernet"

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.7	wioDeviceClass	R	Device class 10 = controller 20 = coupler 30 = switch 40 = display 50 = sensor 60 = actuator
1.3.6.1.4.1.13576.10.1.8	wioDeviceGroup	R	Device group 10 = Serie 750 20 = Serie 758 30 = Serie 767 40 = Serie 762 PERSPECTO

16.2.3 Versions Group

The version group contains information about the hardware/software versions used in the controller.

Table 419: WAGO MIB – Versions Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.10.1	wioFirmwareIndex	R	Index of firmware version
1.3.6.1.4.1.13576.10.1.10.2	wioHardwareIndex	R	Index of hardware version
1.3.6.1.4.1.13576.10.1.10.3	wioFwlIndex	R	Index of software version from firmware loader
1.3.6.1.4.1.13576.10.1.10.4	wioFirmwareVersion	R	Complete firmware string

16.2.4 Real-Time Clock Group

The real-time clock group contains information about the system's real-time clock.

Table 420: WAGO MIB – Real Time Clock Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.11.1	wioRtcDateTime	R/W	Date/time of coupler in UTC as string. For writing date/time use the following string time 11:22:33 date 13-1-2007 Default value: "time xx:xx:xx date xx-xx-xxxx (UTC)"
1.3.6.1.4.1.13576.10.1.11.2	wioRtcTime	R/W	Date/time of coupler in UTC as integer in seconds from 1970-01-01 Default value: "0"
1.3.6.1.4.1.13576.10.1.11.3	wioTimezone	R/W	"Actual time zone of article in hours (-12 - +12) Default value: "0"
1.3.6.1.4.1.13576.10.1.11.4	wioRtcHourMode	R	Hour mode 0 = 12h mode 1 = 24h mode Default value: "0"
1.3.6.1.4.1.13576.10.1.11.5	wioRtcBatteryStatus	R	RTC battery status: 0 = ok 1 = battery empty Default value: "1"
1.3.6.1.4.1.13576.10.1.11.6	wioRtcDayLightSaving	R/W	Time offset of 1 hour: 0 = not offset 1 = offset 1 hour (DayLightSaving) Default value: "0"

16.2.5 Ethernet Group

The Ethernet group contains the settings for the controller on the Ethernet.

Table 421: WAGO MIB – Ethernet Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.12.1	wioEthernetMode	R/W	IP configuration of Ethernet connection: 0 = fix Ip address 1 = dynamic IP address over Bootp 2 = dynamic IP address over DHCP Default value: "1"
1.3.6.1.4.1.13576.10.1.12.2	wioIp	R/W	Actual IP address of coupler
1.3.6.1.4.1.13576.10.1.12.3	wioSubnetMask	R/W	Actual subnet mask of coupler
1.3.6.1.4.1.13576.10.1.12.4	wioGateway	R/W	Actual gateway IP of coupler
1.3.6.1.4.1.13576.10.1.12.5	wioHostname	R/W	Actual host name of coupler
1.3.6.1.4.1.13576.10.1.12.6	wioDomainName	R/W	Actual domain name of coupler
1.3.6.1.4.1.13576.10.1.12.7	wioDnsServer1	R/W	IP address of first DNS server
1.3.6.1.4.1.13576.10.1.12.8	wioDnsServer2	R/W	IP address of second DNS server

16.2.6 Actual Error Group

The actual error group contains information about the last system status/error status.

Table 422: WAGO MIB – Actual Error Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.20.1	wioErrorGroup	R	Error group of last error
1.3.6.1.4.1.13576.10.1.20.2	wioErrorCode	R	Error code of last error
1.3.6.1.4.1.13576.10.1.20.3	wioErrorArgument	R	Error argument of last error
1.3.6.1.4.1.13576.10.1.20.4	wioErrorDescription	R	Error description string

16.2.7 PLC Project Group

The PLC project group contains information about the controller's PLC program.

Table 423: WAGO MIB – PLC Project Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.30.1	wioProjectId	R	ID of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.2	wioProjectDate	R	Date of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.3	wioProjectName	R	Name of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.4	wioProjectTitle	R	Title of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.5	wioProjectVersion	R	Version of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.6	wioProjectAuthor	R	Author of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.7	wioProjectDescription	R	Description of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.8	wioNumberOfIecTasks	R	Number of IEC tasks in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9	wioIecTaskTable	-	
1.3.6.1.4.1.13576.10.1.30.9.1	wioIecTaskEntry	-	
1.3.6.1.4.1.13576.10.1.30.9.1.1	wioIecTaskId	R	ID of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.2	wioIecTaskName	R	Name of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.3	wioIecTaskStatus	R	Status of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.4	wioIecTaskMode	R	Mode of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.5	wioIecTaskPriority	R	Priority of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.6	wioIecTaskInterval	R	Interval of cyclic IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.7	wioIecTaskEvent	R	Event for IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.8	wioIecTaskCycleCount	R	Count of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.9	wioIecTaskCycleTime	R	Last cycle time of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.10	wioIecTaskCycleTime-Min	R	Minimal cycle time of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.11	wioIecTaskCycleTime-Max	R	Maximal cycle time of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.12	wioIecTaskCycleTime-Avg	R	Average cycle time of IE task in the CoDeSys project

16.2.8 Http Group

The Http group contains information and settings for the controller's Web server.

Table 424: WAGO MIB – Http Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.1.1	wioHttpEnable	R/W	Enable/disable the port of the webserver: 0 = port of webserver disable 1 = port of webserver enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.1.2	wioHttpAuthenticationEnable	R/W	Enable/disable the authentication on the websides: 0 = authentication disabled 1 = authentication enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.1.3	wioHttpPort	R/W	Port of the http web server Default value: { 80 }

16.2.9 Ftp Group

The Ftp group contains information and settings for the controller's Ftp server.

Table 425: WAGO MIB – Ftp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.2.1	wioFtpEnable	R/W	Enable/disable the port of the ftp server 0 = port of ftp server disable 1 = port of ftp server enable Default value: { 1 }

16.2.10 Sntp Group

The Sntp group contains information and settings for the controller's Sntp server.

Table 426: WAGO MIB – Sntp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.3.1	wioSntpEnable	R/W	Enable/disable the port of the SNTP server 0 = port of SNTP server disable 1 = port of SNTP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.3.2	wioSntpServer-Address	R/W	IP address of SNTP server Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.3.3	wioSntpClient-Intervall	R/W	Interval to pool SNTP manager Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.3.4	wioSntpClient-Timeout	R/W	Timeout to corrupt SNTP answer Default value: { 2000 }
1.3.6.1.4.1.13576.10.1.40.3.5	wioSntpClient-DayLightSaving	R/W	Time offset of 1 hour: 0 = not offset 1 = offset 1 hour (DayLightSaving) Default value: { 0 }

16.2.11 Snmp Group

The Snmp group contains information and settings for the controller's SNMP agent.

Table 427: WAGO MIB – Snmp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.1	wioSnmpEnable	R/W	Enable/disable the port of the SNMP server 0 = port of SNMP server disable 1 = port of SNMP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.2.1	wioSnmp1-ProtocolEnable	R/W	Enable/disable first SNMPv1/v2c agent Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.2.2	wioSnmp1-ManagerIp	R/W	IP address of first SNMP server Default value: { 'C0A80101'h }
1.3.6.1.4.1.13576.10.1.40.4.2.3	wioSnmp1-Community	R/W	Community identification string for SNMPv1/v2c Default value: { "public" }
1.3.6.1.4.1.13576.10.1.40.4.2.4	wioSnmp1Trap-V1enable	R/W	Enable/disable SNMPv1 traps to first SNMP server Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.2.5	wioSnmp1Trap-V2enable	R/W	Enable/disable SNMPv2c traps to first SNMP server Default value: { 0 }

Table 427: WAGO MIB – Snmp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.2.6	wioSnmp2-ProtocolEnable	R/W	Enable/disable first SNMPv1/v2c agent Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.2.7	wioSnmp2-ManagerIp	R/W	IP address of second SNMP server Default value: { '00000000'h }
1.3.6.1.4.1.13576.10.1.40.4.2.8	wioSnmp2-Community	R/W	Community identification string for SNMPv1/v2c Default value: { "public" }
1.3.6.1.4.1.13576.10.1.40.4.2.9	wioSnmp2Trap-V1enable	R/W	Enable/disable SNMPv1 traps to first SNMP server Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.2.10	wioSnmp2Trap-V2enable	R/W	Enable/disable SNMPv2c traps to first SNMP server Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.3.1	wioSnmp1User-Enable	R/W	Enable/disable first SNMPv3 user Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.2	wioSnmp1-Authentication-Typ	R/W	Authentication typ for first SNMPv3 user: 0 = no Authentication 1 = MD5 Authentication 2 = SHA1 Authentication Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.3	wioSnmp1-Authentication-Name	R/W	Authentication name for first SNMPv3 user Default value: { "SecurityName" }
1.3.6.1.4.1.13576.10.1.40.4.3.4	wioSnmp1-Authentication-Key	R/W	Authentication key for first SNMPv3 user Default value: { "AuthenticationKey" }
1.3.6.1.4.1.13576.10.1.40.4.3.5	wioSnmp1-PrivacyEnable	R/W	Disable/enable data encryption for first SNMPv3 user: 0 = no Encryption 1 = DES Encryption Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.6	wioSnmp1-PrivacyKey	R/W	Privacy key for SNMPv3 for first SNMPv3 user Default value: { "PrivacyKey" }
1.3.6.1.4.1.13576.10.1.40.4.3.7	wioSnmp1-Notification-Enable	R/W	Enable/disable notification (SNMPv3 traps) with SNMPv3 user Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.8	wioSnmp1-Notification-ReceiverIP	R/W	Receiver IP address for notification (SNMPv3 traps) with SNMPv3 user Default value: { 'C0A80101'h }
1.3.6.1.4.1.13576.10.1.40.4.3.9	wioSnmp2User-Enable	R/W	Enable/disable second SNMPv3 user Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.3.10	wioSnmp2-Authentication-Typ	R/W	Authentication typ for second SNMPv3 user: 0 = no authentication 1 = MD5 authentication 2 = SHA1 authentication Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.11	wioSnmp2-Authentication-Name	R/W	Authentication name for second SNMPv3 user Default value: { "SecurityName" }

Table 427: WAGO MIB – Snmp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.3.12	wioSnmp2-Authentication-Key	R/W	Authentication key for second SNMPv3 user Default value: {"AuthenticationKey"}
1.3.6.1.4.1.13576.10.1.40.4.3.13	wioSnmp2-PrivacyEnable	R/W	Privacy key for SNMPv3 for second SNMPv3 user Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.14	wioSnmp2-PrivacyKey	R/W	Privacy key for SNMPv3 for second SNMPv3 user Default value: {"PrivacyKey" }
1.3.6.1.4.1.13576.10.1.40.4.3.15	wioSnmp2-Notification-Enable	R/W	Enable/disable notification (SNMPv3 traps) with SNMPv3 user Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.3.16	wioSnmp2-Notification-ReceiverIP	R/W	Receiver IP address for notification (SNMPv3 traps) with SNMPv3 user Default value: { '00000000'h }

16.2.12 Snmp Trap String Group

The Snmp trap string group contains strings that are attached to the manufacturer-specific traps.

Table 428: WAGO MIB – Snmp Trap String Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.4.1	wioTrapKbus-Error	R/W	String for 1st SNMP trap Default value: {"Kbus Error" }
1.3.6.1.4.1.13576.10.1.40.4.4.2	wioTrapPlcStart	R/W	String for 2nd SNMP trap Default value: {"Plc Start" }
1.3.6.1.4.1.13576.10.1.40.4.4.3	wioTrapPlcStop	R/W	String for 3rd SNMP trap Default value: {"Plc Stop" }
1.3.6.1.4.1.13576.10.1.40.4.4.4	wioTrapPlc-Reset	R/W	String for 4th SNMP trap Default value: {"Plc Reset" }
1.3.6.1.4.1.13576.10.1.40.4.4.5	wioTrapPlcSoftwareWatchdog	R/W	String for 5th SNMP trap Default value: {"Plc Software Watchdog" }
1.3.6.1.4.1.13576.10.1.40.4.4.6	wioTrapPlc-DivideByZero	R/W	String for 6th SNMP trap Default value: {"Plc Divide By Zero"}
1.3.6.1.4.1.13576.10.1.40.4.4.7	wioTrapPlc-OnlineChange	R/W	String for 7th SNMP trap Default value: {"Plc Online Change"}
1.3.6.1.4.1.13576.10.1.40.4.4.8	wioTrapPlc-Download	R/W	String for 8th SNMP trap Default value: {"Plc Download Programm" }
1.3.6.1.4.1.13576.10.1.40.4.4.9	wioTrapPlc-Login	R/W	String for 9th SNMP trap Default value: {"Plc Login" }
1.3.6.1.4.1.13576.10.1.40.4.4.10	wioTrapPlc-Logout	R/W	String for 10th SNMP trap Default value: {"Plc Logout"}

16.2.13 Snmp User Trap String Group

The Snmp user trap string group contains strings that can be attached to user-specific traps. These strings can be changed via SNMP or Wago_SNMP.lib in CoDeSys.

Table 429: WAGO MIB – Snmp User Trap String Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.5.1	wioUserTrapMsg1	R/W	String for 1st SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.2	wioUserTrapMsg2	R/W	String for 2nd SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.3	wioUserTrapMsg3	R/W	String for 3rd SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.4	wioUserTrapMsg4	R/W	String for 4th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.5	wioUserTrapMsg5	R/W	String for 5th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.6	wioUserTrapMsg6	R/W	String for 6th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.7	wioUserTrapMsg7	R/W	String for 7th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.8	wioUserTrapMsg8	R/W	String for 8th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.9	wioUserTrapMsg9	R/W	String for 9th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.10	wioUserTrapMsg10	R/W	String for 10th SNMP trap

16.2.14 Plc Connection Group

Activate or deactivate the connection to CoDeSys with the Plc connection group.

Table 430: WAGO MIB – Plc Connection Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.5.1	wioCoDeSysEnable	R/W	Enable/disable the port of the CoDeSys server 0 = port of CoDeSys server disable 1 = port of CoDeSys server enable Default value: { 1 }

16.2.15 Modbus Group

The Modbus group contains information and settings about the controller's modbus server.

Table 431: WAGO MIB – Modbus Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.6.1	wioModbusTcp-Enable	R/W	Enable/disable the port of the Modbus TCP server 0 = port of Modbus TCP server disable 1 = port of Modbus TCP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.6.2	wioModbusUdb-Enable	R/W	Enable/disable the port of the Modbus UDP server 0 = port of Modbus UDP server disable 1 = port of Modbus UDP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.6.3	wioMax-Connections	R/W	The maximal count of modbus connections Default value: { 15 }
1.3.6.1.4.1.13576.10.1.40.6.4	wioConnection-Timeout	R/W	Timeout of the modbus connection Default value: { 600 }
1.3.6.1.4.1.13576.10.1.40.6.5	wioModbus-WatchdogMode	R/W	Mode of the modbus watchdog Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.6.6	wioModbus-WatchdogTime	R/W	Timeout of the modbus watchdog Default value: { 100 }
1.3.6.1.4.1.13576.10.1.40.6.7	wioFreeModbus-Sockets	R/W	Unused and free modbus connections Default value: { 15 }
1.3.6.1.4.1.13576.10.1.40.6.8	wioModbus-ConnectionTable	-	
1.3.6.1.4.1.13576.10.1.40.6.8.1	wioModbus-ConnectionEntry	-	
1.3.6.1.4.1.13576.10.1.40.6.8.1.1	wioModbus-ConnectionIndex	R/W	Index of modbus connection
1.3.6.1.4.1.13576.10.1.40.6.8.1.2	wioModbus-ConnectionIp	R/W	IP address of modbus connection
1.3.6.1.4.1.13576.10.1.40.6.8.1.3	wioModbus-ConnectionPort	R/W	Port of modbus connection

16.2.16 Ethernet IP Group

The Ethernet IP group contains information and settings for the controller's Ethernet IP.

Table 432: WAGO MIB – Ethernet IP Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.7.1	wioEthernetIpEnable	R/W	Enable/disable the port of the Ethernet IP server 0 = port of Ethernet IP server disable 1 = port of Ethernet IP server enable Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.7.2	wioEthernetIpVariables-InputCount	R/W	
1.3.6.1.4.1.13576.10.1.40.7.3	wioEthernetIpVariables-OutputCount	R/W	
1.3.6.1.4.1.13576.10.1.40.7.4	wioEthernetIpVariables-PlcInputCount	R/W	
1.3.6.1.4.1.13576.10.1.40.7.5	wioEthernetIpVariables-PlcInputOffset	R/W	
1.3.6.1.4.1.13576.10.1.40.7.6	wioEthernetIpVariables-PlcOutputCount	R/W	
1.3.6.1.4.1.13576.10.1.40.7.7	wioEthernetIpVariables-PlcOutputOffset	R/W	
1.3.6.1.4.1.13576.10.1.40.7.8	wioEthernetIpRunIdle-HeaderOriginatorToTarget	R/W	
1.3.6.1.4.1.13576.10.1.40.7.9	wioEthernetIpRunIdle-HeaderTargetToOriginator	R/W	

16.2.17 Process Image Group

The process image group contains a list of information about the terminals connected to the controller.

Table 433: WAGO MIB – Process Image Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.50.1	wioModulCount	R	Count of modules
1.3.6.1.4.1.13576.10.1.50.2	wioAnalogOutLength	R	Length of analog output process datas
1.3.6.1.4.1.13576.10.1.50.3	wioAnalogInLength	R	Length of analog input process datas
1.3.6.1.4.1.13576.10.1.50.4	wioDigitalOutLength	R	Length of digital output process datas
1.3.6.1.4.1.13576.10.1.50.5	wioDigitalInLength	R	Length of digital input process datas
1.3.6.1.4.1.13576.10.1.50.6	wioDigitalOutOffset	R	Offset of digital output process datas
1.3.6.1.4.1.13576.10.1.50.7	wioDigitalInOffset	R	Offset of digital input process datas
1.3.6.1.4.1.13576.10.1.50.8	wioModuleTable	-	
1.3.6.1.4.1.13576.10.1.50.8.1	wioModuleEntry	-	
1.3.6.1.4.1.13576.10.1.50.8.1.1	wioModuleNumber	R	Number of module slot
1.3.6.1.4.1.13576.10.1.50.8.1.2	wioModuleName	R	Name of module

Table 433: WAGO MIB – Process Image Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.50.8.1.3	wioModuleType	R	Type of module
1.3.6.1.4.1.13576.10.1.50.8.1.4	wioModuleCount	R	Count of module
1.3.6.1.4.1.13576.10.1.50.8.1.5	wioModule-AlternativeFormat	R	Module in alternative format
1.3.6.1.4.1.13576.10.1.50.8.1.6	wioModuleAnalog-OutLength	R	Length of analog output data of module (Bit)
1.3.6.1.4.1.13576.10.1.50.8.1.7	wioModuleAnalog-InLength	R	Length of analog input data of module (Bit)
1.3.6.1.4.1.13576.10.1.50.8.1.8	wioModuleDigital-OutLength	R	Length of digital output data of module (Bit)
1.3.6.1.4.1.13576.10.1.50.8.1.9	wioModuleDigital-InLength	R	Length of digital input data of module (Bit)

16.2.18 Plc Data Group

The Plc data group contains values that can be used for data exchange with CoDeSys.

Table 434: WAGO MIB – Plc Data Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.100.1	wioPlcDataTable	-	
1.3.6.1.4.1.13576.10.1.100.1.1	wioPlcDataEntry	-	
1.3.6.1.4.1.13576.10.1.100.1.1.1	wioPlcDataIndex	R/W	Number of plc data DWORD
1.3.6.1.4.1.13576.10.1.100.1.1.2	wioPlcDataReadArea	R/W	Readable plc data (DWORD)
1.3.6.1.4.1.13576.10.1.100.1.1.3	wioPlcDataWriteArea	R	Write-/readable plc data (DWORD)

List of Figures

Figure 1: Fieldbus node.....	21
Figure 2: Example of a manufacturing number.....	22
Figure 3: Isolation for Standard Couplers/Controllers and extended ECO Couplers	24
Figure 4: System supply for standard coupler/controller and extended ECO couplers.....	25
Figure 5: System voltage for standard couplers/controllers and extended ECO couplers.....	26
Figure 6: Field supply (sensor/actuator) for standard couplers/controllers and extended ECO couplers.....	29
Figure 7: Supply module with fuse carrier (Example 750-610).....	31
Figure 8: Removing the fuse carrier.....	31
Figure 9: Opening the fuse carrier.....	32
Figure 10: Change fuse.....	32
Figure 11: Fuse modules for automotive fuses, series 282.....	33
Figure 12: Fuse modules for automotive fuses, series 2006.....	33
Figure 13: Fuse modules with pivotable fuse carrier, series 281.....	33
Figure 14: Fuse modules with pivotable fuse carrier, series 2002.....	33
Figure 15: Power supply concept.....	34
Figure 16: Supply example for standard couplers/controllers and extended ECO couplers.....	36
Figure 17: Carrier rail contact.....	39
Figure 18: Ring-feeding.....	40
Figure 19: Example of the WAGO shield connecting system.....	42
Figure 20: Application of the WAGO shield connecting system.....	42
Figure 21: View ETHERNET TCP/IP Fieldbus Controller.....	46
Figure 22: Device Supply.....	48
Figure 23: RJ-45-Connector.....	49
Figure 24: Display Elements.....	50
Fig. 25: Service interface for programming and configuration (closed and open door).....	51
Figure 26: Mode selector switch (closed and open damper of the service port)...	52
Figure 27: Address Selection Switch.....	54
Figure 28: Spacing.....	63
Figure 29: Unlocking lug of extended ECO coupler.....	66
Figure 30: Insert I/O module.....	67
Figure 31: Snap the I/O module into place.....	67
Figure 32: Removing the I/O module.....	68
Figure 33: Data contacts.....	69
Figure 34: Example for the arrangement of power contacts.....	70
Figure 35: Connecting a conductor to a CAGE CLAMP®.....	71
Figure 36: Run-up of the Controller.....	73
Figure 37: Example of process image for input data.....	76
Figure 38: Example of process image for output data.....	77
Figure 39: Memory areas and data exchange.....	81
Figure 40: Example declaration of remanent flags by „var retain“.....	83
Figure 41: WAGO I/O Configurator.....	84
Figure 42: Data exchange between MODBUS Master and I/O modules.....	92

Figure 43: Data exchange between PLC function (CPU) of the PFC and the I/O modules	94
Figure 44: Example of addressing for a fieldbus node.....	97
Figure 45: Address selection switch.....	101
Figure 46: WBM page "Port"	104
Figure 47: WBM page "Information"	109
Figure 48: WBM page "Port"	110
Figure 49: Example for the Function test of a Fieldbus Node	112
Figure 50: Example of real-time clock synchronization in ETHERNET Settings	116
Figure 51: Dialog window for target system settings.....	119
Figure 52: Write access via module parameters.....	123
Figure 53: EA-config.xml	124
Figure 54: Watchdog runtime is less than the task runtime	132
Figure 55: Watchdog runtime is greater than the task call interval.....	132
Figure 56: System events	135
Figure 57: Dialog window "Communication parameters"	139
Figure 58: WBM page "Information"	144
Figure 59: WBM-Seite „Ethernet“	146
Figure 60: WBM page "TCP/IP".....	149
Figure 61: WBM page "Port"	151
Figure 62: WBM page "SNMP".....	154
Figure 63: WBM page "SNMP V3".....	156
Figure 64: WBM page "Watchdog"	158
Figure 65: WBM page "Clock"	161
Figure 66: WBM page "Security"	163
Figure 67: WBM page „PLC Info“	165
Figure 68: WBM page "PLC"	166
Figure 69: WBM page "Features"	169
Figure 70: WBM page "I/O Config"	170
Figure 71: WBM page "WebVisu"	173
Figure 72: Display Elements	174
Figure 73: Node status - I/O LED signaling.....	177
Figure 74: Error message coding.....	177
Figure 75: Function block for determining loss of fieldbus, independently of protocol	185
Figure 76: Use of the MODBUS Functions	207
Figure 77: Example SCADA software with MODBUS driver	354
Figure 78: Side marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC	356
Figure 79: Printing Text detail – Marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC.....	356
Figure 80: Side marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC	357
Figure 81: Text detail – Marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC.....	357
Figure 82: Side marking example for I/O modules according to NEC 500	359
Figure 83: Text detail – Marking example for I/O modules according to NEC 500	359

List of Tables

Table 1: Number Notation.....	15
Table 2: Font Conventions	15
Table 3: Alignment.....	26
Table 4: Power supply modules	30
Table 5: Filter modules for 24-volt supply.....	34
Table 6: WAGO Power Supply Unit.....	37
Table 7: WAGO ground wire terminals	38
Table 8: Compatibility.....	45
Table 9: Legend to the View ETHERNET TCP/IP Fieldbus Controller	47
Table 10: RJ-45 Connector and RJ-45 Connector Configuration	49
Table 11: Display Elements Fieldbus Status	50
Table 12: Display Elements Node Status	50
Table 13: Display Elements Supply Voltage	50
Table 14: Service port	51
Table 15: Mode selector switch.....	52
Table 16: Meaning of DIP switch positions.....	54
Table 17: Technical data – Device data	55
Table 18: Technical data – System data.....	55
Table 19: Technical data – Supply	56
Table 20: Technical data – Fieldbus MODBUS/TCP	56
Table 21: Technical data – Accessories	56
Table 22: Technical Data Wire Connection.....	56
Table 23: Technical Data - Climatic environmental conditions.....	57
Table 24: Technical data – Mechanical strength.....	57
Table 25: WAGO DIN Rail.....	63
Table 26: Data with for I/O modules.....	85
Table 27: Breakdown of address range	86
Table 28: Address range Word 0...255.....	86
Table 29: Address range, word 256 – 511.....	86
Table 30: Address range, word 512 - 1275	87
Table 31: Address range, word 1276...1531.....	87
Table 32: Address range for flags	87
Table 33: IEC-61131-3 address areas	88
Table 34: Absolute Addressing	89
Table 35: Addressing example	89
Table 36: Allocation of digital inputs and outputs to process data words in accordance with the.....	91
Table 37: ETHERNET libraries for WAGO-I/O-PRO	126
Table 38: Task processing.....	133
Table 39: WBM page "Information"	145
Table 40: WBM page "Ethernet"	147
Table 41: WBM page „TCP/IP“	150
Table 42: WBM page "Port"	152
Table 43: WBM page "SNMP"	155
Table 44: WBM page "Watchdog"	159
Table 45: WBM page "Clock"	161
Table 46: WBM page "Security"	164

Table 47: WBM page „PLC Info“	165
Table 48: WBM page "PLC"	167
Table 49: WBM page "Features"	169
Table 50: WBM page "I/O configuration"	171
Table 51: LED assignment for diagnostics	174
Table 52: Fieldbus diagnostics – solution in event of error	175
Table 53: Node status diagnostics – solution in event of error	176
Table 54: Blink code- table for the I/O LED signaling, error code 1.....	178
Table 55: Blink code table for the I/O LED signaling, error code 2.....	179
Table 56: Blink code table for the I/O LED signaling, error code 3.....	180
Table 57: Blink code table for the I/O LED signaling, error code 4.....	181
Table 58: Blink code table for the I/O LED signaling, error code 5.....	181
Table 59: Blink code- table for the I/O LED signaling, error code 6.....	182
Table 60: Blink code table for the 'I/O' LED signaling, error code 7...9	182
Table 61: Blink code table for the 'I/O' LED signaling, error code 10	183
Table 62: Blink code table for the 'I/O' LED signaling, error code 11	183
Table 63: Power supply status diagnostics – solution in event of error	184
Table 64: IP Packet.....	188
Table 65: Network Class A	189
Table 66: Network Class B	189
Table 67: Network Class C	189
Table 68: Key Data Class A, B and C	190
Table 69: Class B Address with Field for Subnet IDs	191
Table 70: Subnet mask for Class A network.....	191
Table 71: Subnet mask for Class B network.....	191
Table 72: Subnet mask for Class C network.....	191
Table 73: Example for an IP address from a Class B network.....	191
Table 74: BootP options	195
Table 75: Meaning of DHCP options.....	198
Table 76: Meaning of the SNTP Parameters.....	199
Table 77: MIB II groups.....	201
Table 78: Standard Traps	202
Table 79: MODBUS/TCP header.....	204
Table 80: Basic data types of MODBUS protocol	205
Table 81: List of the MODBUS functions in the fieldbus controller.....	206
Table 82: Exception Codes.....	208
Table 83: Request of Function code FC1	209
Table 84: Response of Function code FC1	209
Table 85: Assignment of inputs.....	209
Table 86: Exception of Function code FC1.....	210
Table 87: Request of Function code FC2.....	211
Table 88: Response of Function code FC2	211
Table 89: Assignment of inputs.....	211
Table 90: Exception of Function code FC2.....	212
Table 91: Request of Function code FC3	213
Table 92: Response of Function code FC3	213
Table 93: Exception of Function code FC3.....	213
Table 94: Request of Function code FC4.....	214
Table 95: Response of Function code FC4	214
Table 96: Exception of Function code FC4.....	214

Table 97: Request of Function code FC5	215
Table 98: Response of Function code FC5	215
Table 99: Exception of Function code FC5	215
Table 100: Request of Function code FC6	216
Table 101: Response of Function code FC6	216
Table 102: Exception of Function code FC6	216
Table 103: Request of Function code FC11	217
Table 104: Response of Function code FC11	217
Table 105: Exception of Function code FC 11	217
Table 106: Request of Function code FC15	218
Table 107: Response of Function code FC15	218
Table 108: Exception of Function code FC15	219
Table 109: Request of Function code FC16	220
Table 110: Response of Function code FC16	220
Table 111: Exception of Function code FC16	220
Table 112: Request of Function code FC22	221
Table 113: Response of Function code FC22	221
Table 114: Exception of Function code FC22	221
Table 115: Request of Function code FC23	222
Table 116: Response of Function code FC23	222
Table 117: Exception of Function code FC23	222
Table 118: Register access reading (with FC3, FC4 and FC23)	224
Table 119: Register access writing (with FC6, FC16, FC22 and FC23)	225
Table 120: Bit access reading (with FC1 and FC2)	226
Table 121: Bit access writing (with FC5 and FC15)	226
Table 122: MODBUS registers	227
Table 123: MODBUS registers (Continuation)	228
Table 124: Register address 0x1000	229
Table 125: Register address 0x1001	229
Table 126: Register address 0x1002	230
Table 127: Register address 0x1003	230
Table 128: Register address 0x1004	230
Table 129: Register address 0x1005	230
Table 130: Register address 0x1006	231
Table 131: Register address 0x1007	231
Table 132: Register address 0x1008	231
Table 133: Register address 0x1009	231
Table 134: Register address 0x100A	231
Table 135: Starting Watchdog	232
Table 136: Register address 0x100B	233
Table 137: Register address 0x1020	234
Table 138: Register address 0x1021	234
Table 139: Register address 0x1022	235
Table 140: Register address 0x1023	235
Table 141: Register address 0x1024	235
Table 142: Register address 0x1025	235
Table 143: Register address 0x1028	235
Table 144: Register address 0x1029	236
Table 145: Register address 0x102A	236
Table 146: Register address 0x102B	236

Table 147: Register address 0x1030	236
Table 148: Register address 0x1031	236
Table 149: Register address 0x1050	237
Table 150: Register address 0x2030	238
Table 151: Register address 0x2031	238
Table 152: Register address 0x2032	238
Table 153: Register address 0x2033	239
Table 154: Register address 0x2040	239
Table 155: Register address 0x2041	239
Table 156: Register address 0x2042	239
Table 157: Register address 0x2043	239
Table 158: Register address 0x2010	240
Table 159: Register address 0x2011	240
Table 160: Register address 0x2012	240
Table 161: Register address 0x2013	240
Table 162: Register address 0x2014	240
Table 163: Register address 0x2020	240
Table 164: Register address 0x2021	241
Table 165: Register address 0x2022	241
Table 166: Register address 0x2023	241
Table 167: Register address 0x2000	242
Table 168: Register address 0x2001	242
Table 169: Register address 0x2002	242
Table 170: Register address 0x2003	242
Table 171: Register address 0x2004	242
Table 172: Register address 0x2005	242
Table 173: Register address 0x2006	243
Table 174: Register address 0x2007	243
Table 175: Register address 0x2008	243
Table 176: Register address 0x3000 to 0x5FFF	243
Table 177: ISO/OSI reference model	245
Table 178: CIP common class	248
Table 179: WAGO specific classes	249
Table 180: Explanation of the table headings in the object descriptions	250
Table 181: Identity (01 _{hex}) – Class	250
Table 182: Identity (01 _{hex}) – Instance 1	251
Table 183: Identity (01 _{hex}) – Common service	251
Table 184: Message router (02 _{hex}) – Class	252
Table 185: Message router (02 _{hex}) – Instance 1	252
Table 186: Message router (02 _{hex}) – Common service	253
Table 187: Assembly (04 _{hex}) – Class	253
Table 188: Overview of static Assembly instances	253
Table 189: Static assembly instances – Instance 101 (65 _{hex})	254
Table 190: Static assembly instances – Instance 102 (66 _{hex})	254
Table 191: Static assembly instances – Instance 103 (67 _{hex})	254
Table 192: Static assembly instances – Instance 104 (68 _{hex})	254
Table 193: Static assembly instances – Instance 105 (69 _{hex})	255
Table 194: Static assembly instances – Instance 106 (6A _{hex})	255
Table 195: Static assembly instances – Instance 107 (6B _{hex})	255
Table 196: Static assembly instances – Instance 108 (6C _{hex})	256

Table 197: Static assembly instances – Instance 109 (6C _{hex}).....	256
Table 198: Static assembly instances – Instance 110 (6E _{hex}).....	256
Table 199: Static assembly instances – Instance 111 (6F _{hex}).....	256
Table 200: Static assembly instances – Common service.....	257
Table 201: Port class (F4 _{hex}) – Class.....	258
Table 202: Port class (F4 _{hex}) – Instance 1.....	258
Table 203: Port class (F4 _{hex}) – Common service.....	259
Table 204: TCP/IP interface (F5 _{hex}) – Class.....	260
Table 205: TCP/IP interface (F5 _{hex}) – Instance 1.....	261
Table 206: TCP/IP interface (F5 _{hex}) – Common service.....	261
Table 207: Ethernet link (F5 _{hex}) – Class.....	262
Table 208: Ethernet link (F6 _{hex}) – Instance 1.....	263
Table 209: Ethernet link (F6 _{hex}) – Instance 2.....	265
Table 210: Ethernet link (F6 _{hex}) – Instance 3.....	267
Table 211: Ethernet link (F6 _{hex}) – Common service.....	267
Table 212: Coupler/Controller configuration (64 _{hex}) – Class.....	268
Table 213: Coupler/Controller configuration (64 _{hex}) – Instance 1.....	268
Table 214: Coupler/Controller configuration (64 _{hex}) – Common service.....	270
Table 215: Discrete input point (65 _{hex}) – Class.....	270
Table 216: Discrete input point (65 _{hex}) – Instance 1...255.....	270
Table 217: Discrete input point (65 _{hex}) – Common service.....	270
Table 218: Discrete Input Point Extended 1(69 _{hex}) – Class.....	271
Table 219: Discrete output point (66 _{hex}) – Instance 256...510.....	271
Table 220: Discrete Input Point Extended 1 (69 _{hex}) – Common service.....	271
Table 221: Discrete Input Point Extended 2 (6D _{hex}) – Class.....	271
Table 222: Analog input point (67 _{hex}) – Instance 1.....	271
Table 223: Analog input point (67 _{hex}) – Common service.....	272
Table 224: Discrete Input Point Extended 3 (71 _{hex}) – Class.....	272
Table 225: Discrete Input Point Extended 3 (71 _{hex}) – Instance 766...1020.....	272
Table 226: Discrete Input Point Extended 3 (71 _{hex}) – Common service.....	272
Table 227: Discrete Output Point (66 _{hex}) – Class.....	272
Table 228: Discrete Output Point (66 _{hex}) – Instance 1...255.....	273
Table 229: Discrete Output Point (66 _{hex}) – Common service.....	273
Table 230: Discrete Output Point Extended 1 (6A _{hex}) – Class.....	273
Table 231: Discrete Output Point Extended 1 (6A _{hex}) – Instance 256...510.....	273
Table 232: Discrete Output Point Extended 1 (6A _{hex}) – Common service.....	273
Table 233: Discrete Output Point Extended 2 (6E _{hex}) – Class.....	274
Table 234: Discrete Output Point Extended 2 (6E _{hex}) – Instance 511...765.....	274
Table 235: Discrete Output Point Extended 2 (6E _{hex}) – Common service.....	274
Table 236: Discrete Output Point Extended 3 (72 _{hex}) – Class.....	274
Table 237: Discrete Output Point Extended 3 (72 _{hex}) – Instance 766...1020.....	275
Table 238: Discrete Output Point Extended 2 (6E _{hex}) – Common service.....	275
Table 239: Analog Input Point (67 _{hex}) – Class.....	275
Table 240: Analog Input Point (67 _{hex}) – Instance 1 ... 255.....	275
Table 241: Analog Input Point (67 _{hex}) – Common service.....	275
Table 242: Analog Input Point Extended 1 (6B _{hex}) – Class.....	276
Table 243: Analog Input Point Extended 1 (6B _{hex}) – Instance 256 ... 510.....	276
Table 244: Analog Input Point Extended 1 (6B _{hex}) – Common service.....	276
Table 245: Analog Input Point Extended 2 (6F _{hex}) – Class.....	276
Table 246: Analog Input Point Extended 2 (6F _{hex}) – Instance 511 ... 765.....	277

Table 247: Analog Input Point Extended 2 (6F _{hex}) – Common service	277
Table 248: Analog Input Point Extended 3 (73 _{hex}) – Class	277
Table 249: Analog Input Point Extended 3 (73 _{hex}) – Instance 766 ... 1020	277
Table 250: Analog Input Point Extended 3 (73 _{hex}) – Common service	278
Table 251: Analog Output Point (68 _{hex}) – Class	278
Table 252: Analog Output Point (68 _{hex}) – Instance 1...255	278
Table 253: Analog Output Point (68 _{hex}) – Common service	278
Table 254: Analog Output Point Extended 1 (6C _{hex}) – Class	279
Table 255: Analog Output Point Extended 1 (6C _{hex}) – Instance 256...510	279
Table 256: Analog Output Point Extended 1 (6C _{hex}) – Common service	279
Table 257: Analog Output Point Extended 2 (70 _{hex}) – Class	279
Table 258: Analog Output Point Extended 2 (70 _{hex}) – Instance 511...765	280
Table 259: Analog Output Point Extended 2 (70 _{hex}) – Common service	280
Table 260: Analog Output Point Extended 3 (74 _{hex}) – Class	280
Table 261: Analog Output Point Extended 3 (74 _{hex}) – Instance 766...1020	280
Table 262: Analog Output Point Extended 3 (74 _{hex}) – Common service	281
Table 263: Module Configuration (80 _{hex}) – Class	281
Table 264: Module Configuration (80 _{hex}) – Instance 1...255	281
Table 265: Module Configuration (80 _{hex}) – Common service	281
Table 266: Module Configuration Extended (81 _{hex}) – Class	282
Table 267: Module Configuration Extended (81 _{hex}) – Instance 256	282
Table 268: Module Configuration Extended (81 _{hex}) – Common service	282
Table 269: Input fieldbus variable USINT (A0 _{hex}) – Class	283
Table 270: Input fieldbus variable USINT (A0 _{hex}) – Instance 1...255	283
Table 271: Input fieldbus variable USINT (A0 _{hex}) – Common service	283
Table 272: Input Fieldbus Variable USINT Extended 1 (A1 _{hex}) – Class	284
Table 273: Input fieldbus variable USINT Extended 1 (A1 _{hex}) – Instance 256...510	284
Table 274: Input fieldbus variable USINT Extended 1 (A1 _{hex}) – Common service	284
Table 275: Input Fieldbus Variable USINT Extended 2 (A2 _{hex}) – Class	285
Table 276: Input Fieldbus Variable USINT Extended 2 (A2 _{hex}) – Instance 511...512	285
Table 277: Input fieldbus variable USINT Extended 2 (A2 _{hex}) – Common service	285
Table 278: Output fieldbus variable USINT (A3 _{hex}) – Class	286
Table 279: Output fieldbus variable USINT (A3 _{hex}) – Instance 1...255	286
Table 280: Output fieldbus variable USINT (A3 _{hex}) – Common service	286
Table 281: Output Fieldbus variable USINT Extended 1 (A4 _{hex}) – Class	287
Table 282: Output Fieldbus Variable USINT Extended 1 (A4 _{hex}) – Instance 256...510	287
Table 283: Output Fieldbus Variable USINT Extended 1 (A4 _{hex}) – Common service	287
Table 284: Output Fieldbus Variable USINT Extended 2 (A5 _{hex}) – Class	288
Table 285: Output Fieldbus Variable USINT Extended 2 (A5 _{hex}) – Instance 511...512	288
Table 286: Output Fieldbus Variable USINT Extended 2 (A5 _{hex}) – Common service	288
Table 287: Input fieldbus variable UINT (A6 _{hex}) – Class	289
Table 288: Input fieldbus variable UINT (A6 _{hex}) – Instance 1...255	289

Table 289: Input fieldbus variable UINT (A6 _{hex}) – Common service	289
Table 290: Input Fieldbus Variable UINT Extended 1 (A7 _{hex}) – Class	290
Table 291: Input Fieldbus Variable UINT Extended 1 (A7 _{hex}) – Instance 256 ..	290
Table 292: Input Fieldbus Variable UINT Extended 1 (A7 _{hex}) – Common service	290
Table 293: Output fieldbus variable UINT (A8 _{hex}) – Class	291
Table 294: Output fieldbus variable UINT (A8 _{hex}) – Instance 1...255	291
Table 295: Output fieldbus variable UINT (A8 _{hex}) – Common service	291
Table 296: Output Fieldbus Variable UINT Extended 1 (A9 _{hex}) – Class	292
Table 297: Output Fieldbus Variable UINT Extended 1 (A9 _{hex}) – Instance 256	292
Table 298: Output Fieldbus Variable UINT Extended 1 (A9 _{hex}) – Common service	292
Table 299: Input fieldbus variable UDINT (AA _{hex}) – Class	293
Table 300: Input fieldbus variable UDINT (AA _{hex}) – Instance 1...128	293
Table 301: Input fieldbus variable UDINT (AA _{hex}) – Common service	293
Table 302: Input Fieldbus Variable UDINT Offset (AB _{hex}) – Class	294
Table 303: Input Fieldbus Variable UDINT Offset (AB _{hex}) – Instance 1...128 ..	294
Table 304: Input Fieldbus Variable UDINT Offset (AB _{hex}) – Common service	294
Table 305: Input fieldbus variable UDINT (AA _{hex}) – Class	295
Table 306: Input fieldbus variable UDINT (AA _{hex}) – Instance 1...128	295
Table 307: Input fieldbus variable UDINT (AA _{hex}) – Common service	295
Table 308: Output Fieldbus Variable UDINT Offset (AD _{hex}) – Class	296
Table 309: Output Fieldbus Variable UDINT Offset (AD _{hex}) – Instance 1...128	296
Table 310: Output Fieldbus Variable UDINT Offset (AD _{hex}) – Common service	296
Table 311: 1 Channel Digital Input Module with Diagnostics	299
Table 312: 2 Channel Digital Input Modules	299
Table 313: 2 Channel Digital Input Module with Diagnostics	299
Table 314: 2 Channel Digital Input Module with Diagnostics and Output Process Data	300
Table 315: 4 Channel Digital Input Modules	300
Table 316: 8 Channel Digital Input Modules	300
Table 317: 16 Channel Digital Input Modules	301
Table 318: 1 Channel Digital Output Module with Input Process Data	302
Table 319: 2 Channel Digital Output Modules	302
Table 320: 2 Channel Digital Input Modules with Diagnostics and Input Process Data	303
Table 321: 2 Channel Digital Input Modules with Diagnostics and Input Process Data 75x-506	303
Table 322: 4 Channel Digital Output Modules	304
Table 323: 4 Channel Digital Output Modules with Diagnostics and Input Process Data	304
Table 324: 8 Channel Digital Output Module	304
Table 325: 8 Channel Digital Output Modules with Diagnostics and Input Process Data	305
Table 326: 16 Channel Digital Output Modules	305
Table 327: 8 Channel Digital Input/Output Modules	306
Table 328: 1 Channel Analog Input Modules	307
Table 329: 2 Channel Analog Input Modules	307

Table 330: 4 Channel Analog Input Modules	308
Table 331: 2 Channel Analog Output Modules.....	309
Table 332: 4 Channel Analog Output Modules.....	309
Table 333: Counter Modules 750-404, (and all variations except of /000-005), 753-404, (and variation /000-003)	310
Table 334: Counter Modules 750-404/000-005	311
Table 335: Counter Modules 750-638, 753-638	311
Table 336: Pulse Width Modules 750-511, /xxx-xxx	312
Table 337: Serial Interface Modules with alternative Data Format	312
Table 338: Serial Interface Modules with Standard Data Format.....	313
Table 339: Data Exchange Module	313
Table 340: SSI Transmitter Interface Modules	314
Table 341: Incremental Encoder Interface Modules 750-631/000-004, --010, -011	314
Table 342: Incremental Encoder Interface Modules 750-634.....	315
Table 343: Incremental Encoder Interface Modules 750-637.....	315
Table 344: Digital Pulse Interface Modules 750-635	316
Table 345: DC-Drive Controller 750-636.....	316
Table 346: Stepper Controller RS 422 / 24 V / 20 mA 750-670.....	317
Table 347: RTC Module 750-640	318
Table 348: DALI/DSI Master module 750-641	318
Table 349: EnOcean Radio Receiver 750-642.....	319
Table 350: MP Bus Master Module 750-643.....	319
Table 351: Bluetooth® RF-Transceiver 750-644.....	320
Table 352: Vibration Velocity/Bearing Condition Monitoring VIB I/O 750-645	321
Table 353: KNX/EIB/TP1 Module 753-646.....	322
Table 354: AS-interface Master module 750-655.....	323
Table 355: System Modules with Diagnostics 750-610, -611	324
Table 356: Binary Space Module 750-622 (with behavior like 2 channel digital input).....	324
Table 357: 1 Channel Digital Input Module with Diagnostics	326
Table 358: 2 Channel Digital Input Modules.....	326
Table 359: 2 Channel Digital Input Module with Diagnostics	327
Table 360: 2 Channel Digital Input Module with Diagnostics and Output Process Data.....	327
Table 361: 4 Channel Digital Input Modules.....	328
Table 362: 8 Channel Digital Input Modules.....	328
Table 363: 16 Channel Digital Input Modules.....	328
Table 364: 1 Channel Digital Output Module with Input Process Data	329
Table 365: 2 Channel Digital Output Modules	330
Table 366: 2 Channel Digital Input Modules with Diagnostics and Input Process Data.....	330
Table 367: 2 Channel Digital Input Modules with Diagnostics and Input Process Data 75x-506.....	331
Table 368: 4 Channel Digital Output Modules	331
Table 369: 4 Channel Digital Output Modules with Diagnostics and Input Process Data.....	331
Table 370: 8 Channel Digital Output Module.....	332

Table 371: 8 Channel Digital Output Modules with Diagnostics and Input Process Data	332
Table 372: 16 Channel Digital Output Modules	333
Table 373: 8 Channel Digital Input/Output Modules	333
Table 374: 1 Channel Analog Input Modules	334
Table 375: 2 Channel Analog Input Modules	335
Table 376: 4 Channel Analog Input Modules	335
Table 377: 2 Channel Analog Output Modules	336
Table 378: 4 Channel Analog Output Modules	336
Table 379: Counter Modules 750-404, (and all variations except of /000-005), 753-404, (and variation /000-003)	337
Table 380: Counter Modules 750-404/000-005	338
Table 381: Counter Modules 750-638, 753-638	339
Table 382: Pulse Width Modules 750-511, /xxx-xxx	339
Table 383: Serial Interface Modules with alternative Data Format	340
Table 384: Serial Interface Modules with Standard Data Format	340
Table 385: Data Exchange Module	341
Table 386: SSI Transmitter Interface Modules	341
Table 387: SSI Transmitter Interface I/O Modules with an Alternative Data Format	342
Table 388: Incremental Encoder Interface Modules 750-631/000-004, -010, -011	342
Table 389: Incremental Encoder Interface Modules 750-634	343
Table 390: Incremental Encoder Interface Modules 750-637	343
Table 391: Incremental Encoder Interface Modules 750-635, 750-635	344
Table 392: DC-Drive Controller 750-636	344
Table 393: Steppercontroller RS 422 / 24 V / 20 mA 750-670	345
Table 394: RTC Module 750-640	346
Table 395: DALI/DSI Master module 750-641	347
Table 396: EnOcean Radio Receiver 750-642	347
Table 397: MP Bus Master Module 750-643	348
Table 398: Bluetooth® RF-Transceiver 750-644	349
Table 399: Vibration Velocity/Bearing Condition Monitoring VIB I/O 750-645	349
Table 400: AS-interface Master module 750-655	350
Table 401: System Modules with Diagnostics 750-610, -611	351
Table 402: Binary Space Module 750-622 (with behavior like 2 channel digital input)	351
Table 403: MODBUS table and function codes	354
Table 404: Description of marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC	356
Table 405: Description of marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC	358
Table 406: Description of marking example for I/O modules according to NEC 500	359
Table 407: VDE Installation Regulations in Germany	360
Table 408: Installation Regulations in USA and Canada	360
Table 409: MIB II – System group	368
Table 410: MIB II – Interface Group	369
Table 411: MIB II – IP Group	371

Table 412: MIB II – IpRoute Table Group	372
Table 413: MIB II – ICMP Group.....	373
Table 414: MIB II – TCP Group	374
Table 415: MIB II – UDP Group	375
Table 416: MIB II – SNMP Group	376
Table 417: WAGO MIB – Company Group	377
Table 418: WAGO MIB – Product Group	377
Table 419: WAGO MIB – Versions Group	378
Table 420: WAGO MIB – Real Time Clock Group	379
Table 421: WAGO MIB – Ethernet Group	380
Table 422: WAGO MIB – Actual Error Group	380
Table 423: WAGO MIB – PLC Project Group.....	381
Table 424: WAGO MIB – Http Group	382
Table 425: WAGO MIB – Ftp Group	382
Table 426: WAGO MIB – Sntp Group	383
Table 427: WAGO MIB – Snmp Group	383
Table 428: WAGO MIB – Snmp Trap String Group.....	385
Table 429: WAGO MIB – Snmp User Trap String Group	386
Table 430: WAGO MIB – Plc Connection Group	386
Table 431: WAGO MIB – Modbus Group	387
Table 432: WAGO MIB – Ethernet IP Group	388
Table 433: WAGO MIB – Process Image Group	388
Table 434: WAGO MIB – Plc Data Group.....	389

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: +49/5 71/8 87 – 0
Fax: +49/5 71/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>



This Page Left Intentionally Blank



WAGO-I/O-SYSTEM 750 **ETHERNET Programmable Fieldbus** **750-881** **10/100 Mbit/s; digital and analog signals**

Version 1.1.0

WAGO®
INNOVATIVE CONNECTIONS

© 2012 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

1	Notes about this Documentation.....	4
1.1	Validity of this Documentation.....	4
1.2	Copyright.....	4
1.3	Symbols.....	5
1.4	Number Notation.....	7
1.5	Font Conventions	7
2	Important Notes	8
2.1	Legal Bases	8
2.1.1	Subject to Changes	8
2.1.2	Personnel Qualifications.....	8
2.1.3	Use of the 750 Series in Compliance with Underlying Provisions	8
2.1.4	Technical Condition of Specified Devices	9
2.2	Safety Advice (Precautions).....	10
3	Quickstart Description	12
3.1	Hardware Design.....	12
3.2	Preparatory Measures.....	15
3.2.1	Download documentation and sample projects.....	15
3.2.2	Installing WAGO-I/O-PRO.....	15
3.2.3	Starting Up the ETHERNET Network.....	21
3.2.3.1	Determine the IP address of your PC and change it if required	22
3.2.3.2	Assigning an IP address to the fieldbus controller	24
3.2.3.3	Testing the ETHERNET link.....	26
3.2.4	Installing the USB Driver.....	27
4	Application example: Starterkit881_App1_FUP.pro.....	29
4.1	Open project	30
4.2	Configure hardware.....	30
4.3	Main Program "PLC_PRG"	34
4.4	Subprogram "Blink"	35
4.5	Subprogram "UserLED"	37
4.6	Commissioning	41
4.6.1	Configuring a Communication Driver for an ETHERNET Link.....	41
4.7	Configuring a Communication Driver for a USB Link	43
4.7.1	Starting the Program.....	45
4.8	Visualization	46
4.8.1	Opening Visualization via the WAGO-I/O-PRO Software.....	46
4.8.2	Opening Web Visualization in the Internet Browser	47
4.8.3	Operating the Visualization.....	48
	List of Figures	49
	List of Tables.....	51

1 Notes about this Documentation

1.1 Validity of this Documentation

This documentation is only applicable to the 750-881 ETHERNET Programmable Fieldbus of the WAGO-I/O-SYSTEM 750 series.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.3 Symbols

 **DANGER**

Personal Injury!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **DANGER**

Personal Injury Caused by Electric Current!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **WARNING**

Personal Injury!

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

 **CAUTION**

Personal Injury!

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE

Damage to Property Caused by Electrostatic Discharge (ESD)!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note

Important Note!

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.



Information

Additional Information:

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.4 Number Notation

Table 1: Number Notation

Number code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.5 Font Conventions

Table 2: Font Conventions

Font type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Programme\WAGO-I/O-CHECK</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualifications

All sequences implemented on Series 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

All changes to the coupler or controller should always be carried out by qualified personnel with sufficient skills in PLC programming.

2.1.3 Use of the 750 Series in Compliance with Underlying Provisions

Couplers, controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to the actuators or higher-level control systems. Using programmable controllers, the signals can also be (pre-) processed.

The components have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the components in wet and dusty environments is prohibited.

Operating 750 Series components in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section on "WAGO-I/O-SYSTEM 750" → "System Description" → "Technical Data" in the manual for the used fieldbus coupler/controller.

Appropriate housing (per 94/9/EG) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

2.1.4 Technical Condition of Specified Devices

The components to be supplied Ex Works, are equipped with hardware and software configurations, which meet the individual application requirements. WAGO Kontakttechnik GmbH & Co. KG will be exempted from any liability in case of changes in hardware or software as well as to non-compliant usage of components.

Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

2.2 Safety Advice (Precautions)

For installing and operating purposes of the relevant device to your system the following safety precautions shall be observed:



DANGER

Do not work on components while energized!

All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.

DANGER

Installation only in appropriate housings, cabinets or in electrical operation rooms!

The WAGO-I/O-SYSTEM 750 and its components are an open system. As such, install the system and its components exclusively in appropriate housings, cabinets or in electrical operation rooms. Allow access to such equipment and fixtures to authorized, qualified staff only by means of specific keys or tools.

NOTICE

Replace defective or damaged devices!

Replace defective or damaged device/module (e.g., in the event of deformed contacts), since the long-term functionality of device/module involved can no longer be ensured.

NOTICE

Protect the components against materials having seeping and insulating properties!

The components are not resistant to materials having seeping and insulating properties such as: aerosols, silicones and triglycerides (found in some hand creams). If you cannot exclude that such materials will appear in the component environment, then install the components in an enclosure being resistant to the above-mentioned materials. Clean tools and materials are imperative for handling devices/modules.

NOTICE

Cleaning only with permitted materials!

Clean soiled contacts using oil-free compressed air or with ethyl alcohol and leather cloths.

NOTICE

Do not use any contact spray!

Do not use any contact spray. The spray may impair contact area functionality in connection with contamination.

NOTICE

Do not reverse the polarity of connection lines!

Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

NOTICE



Avoid electrostatic discharge!

The devices are equipped with electronic components that you may destroy by electrostatic discharge when you touch. Pay attention while handling the devices to good grounding of the environment (persons, job and packing).

3 Quickstart Description

3.1 Hardware Design

The fieldbus node is to be structured as follows (viewed from left to right):

Table 3: Fieldbus Node Structure

Position	I/O-Modul	
1	PLC ETHERNET	750-881
2	Digital input module	750-400
3	Digital output module	750-501
4	End module	750-600

The 24VDC power supply unit is linked to the power (24V and 0V) for the fieldbus controller and for the power jumper contacts (see No. 5 in the figures of the PLC views and connections below).

For the application used in the example, it is sufficient to connect a jumper between "24V" and "+" (see No. 6) or between "0V" and "-" (see No. 9).

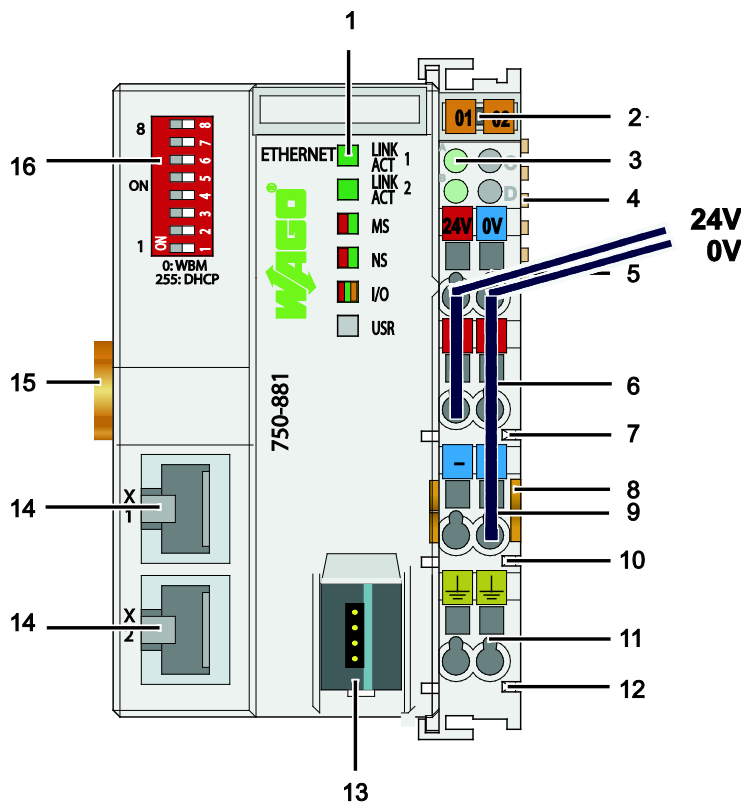


Figure 1: View and 24V/0V connection of the ETHERNET-TCP/IP fieldbus controller

Table 4: Legend to the View ETHERNET TCP/IP Fieldbus Controller

No.	Designation	Meaning	Details see Chapter:
1	LINK ACT 1, 2, MS, NS, I/O, USR	Status LEDs Fieldbus	"Device Description" > "Display Elements"
2	---	Group marking carrier (retractable) with additional marking possibility on two miniature WSB markers	---
3	A, B or C	Status LED's System/Field Supply	"Device Description" > "Display Elements"
4	---	Data Contacts	"Connect Devices" > "Data Contacts/Internal Bus"
5	24 V, 0 V	CAGE CLAMP® Connections System Supply	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
6	+	CAGE CLAMP® Connections Field Supply DC 24 V	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
7	---	Power Jumper Contact 24 V DC	"Connect Devices" > "Power Contacts/ Field Supply"
8	---	Unlocking Lug	"Assembly" > "Inserting and Removing Devices"
9	-	CAGE CLAMP® Connections Field Supply 0 V	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
10	---	Power Jumper Contact 0 V	"Connect Devices" > "Power Contacts/ Field Supply"
11	(Earth)	CAGE CLAMP® Connections Field Supply (Earth)	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
12	---	Power Jumper Contact (Earth)	"Connect Devices" > "Power Contacts/ Field Supply"
13	---	Service Interface (open flap)	"Device Description" > "Operating Elements"
14	X1, X2	Fieldbus connection 2 x RJ-45 as 2-Port ETHERNET Switch	"Device Description" > "Connectors"
15	---	Locking Disc	"Assembly" > "Inserting and Removing Devices"
16	---	Address Selection Switch	"Device Description" > "Operating Elements"

Note



Set the operating mode switch to the top position!

Set the operating mode switch to the top position. At this setting, the firmware and application program are executed (Activate program execution / "RUN").

The mode selector switch is located behind the cover flap (see No. 13 in the Figure above).

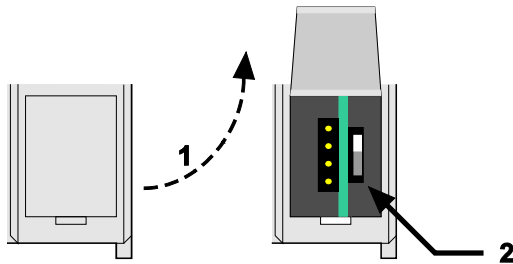


Figure 2: Mode selector switch (closed and open damper of the service port)

Table 5: Mode selector switch

Number	Description
1	Open the damper
2	Operating mode switch

3.2 Preparatory Measures

3.2.1 Download documentation and sample projects



Note

Download documentation and sample projects!

Extensive documentation and sample projects for this Starterkit are available at the WAGO homepage. You will find these at www.wago.com/startkits.

Description	Version	Link
Installation Guide German/English	1.0.0	
Quick-Start Guide German	1.0.0	
Quick-Start Guide English	1.0.1	
German Manual	1.0.0	
English Manual	1.0.0	

Applications

Name	Description	Version	Download
Application 1 (Function Plan)	This application example, including visualization, shows how to control and process digital inputs and outputs.	1.0	
Application 1	This application example, including visualization, shows how to control and process digital inputs and outputs.	Ladder diagram	

Figure 3: Download possibilities at the WAGO homepage

3.2.2 Installing WAGO-I/O-PRO

The Starterkit ETHERNET 750-881 includes a CD with the development topology WAGO-I/O-PRO. This CD is also available individually under item number 759-333.

WAGO-I/O-PRO is a programming and visualization tool for control programs. This software is used to develop PLC applications for WAGO I/O SYSTEM 750 programmable fieldbus controllers. WAGO-I/O-PRO runs in compliance with the

IEC 61131-3 standard, which specifies the requirements for a programming system.

You can use this programming tool to develop application programs in the programming languages AWL, KOP, FUP, CFC, ST and AS.

Proceed as follows to perform the installation:

1. Insert the WAGO-I/O-PRO CD into your CD-ROM drive.
2. If the installation does not begin automatically, start the file Setup.exe.

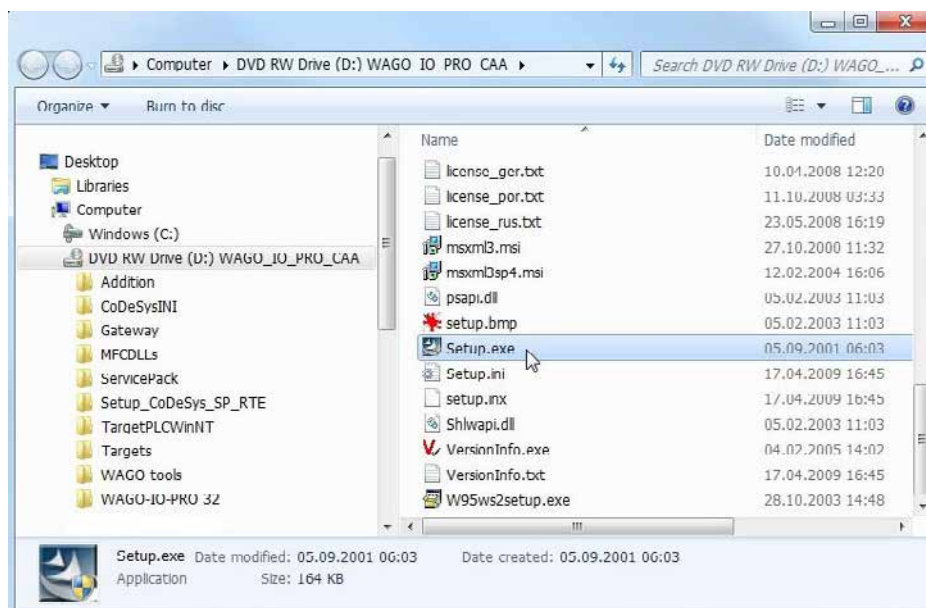


Figure 4: Contents of the WAGO-I/O-PRO CD (folder view), Setup.exe file

3. Select the language. Then confirm this selection by clicking [OK].

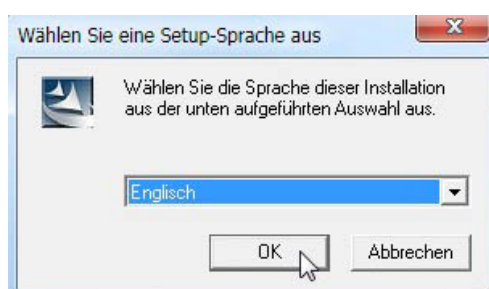


Figure 5: Selecting the setup language

4. Click the **[Next]** button in order to start setup using the "InstallShield Wizard".

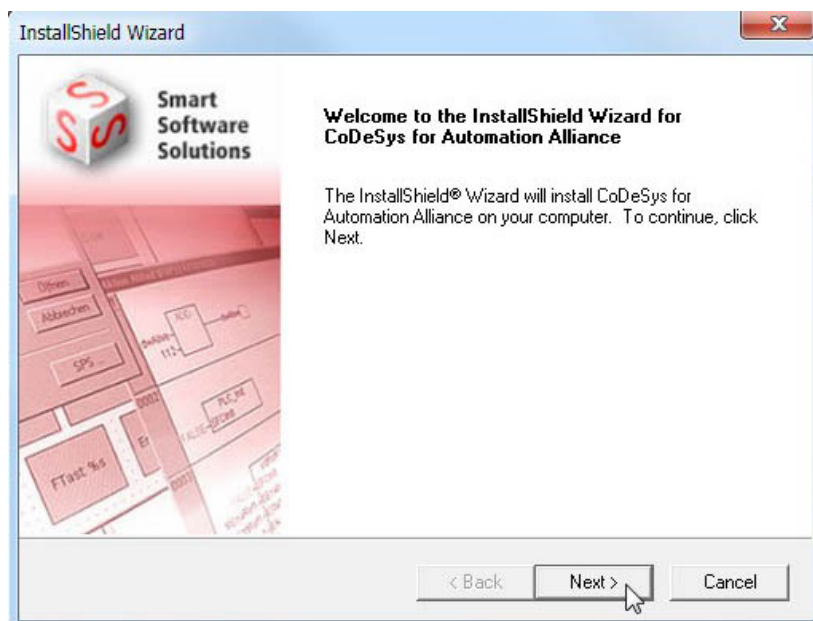


Figure 6: Starting the setup

5. Click on the **[OK]** button in order to accept the license agreement.

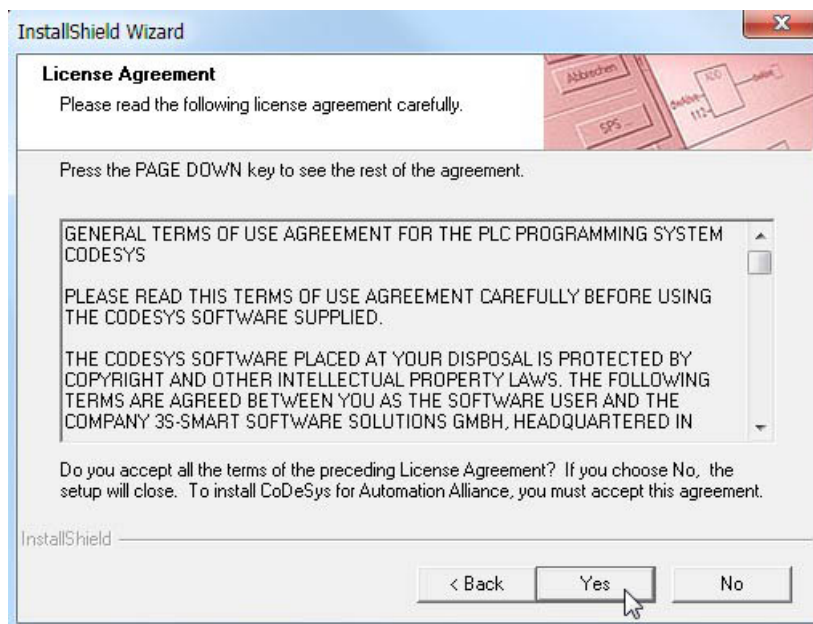


Figure 7: License agreement

6. Select the target directory. Then confirm this selection by clicking **[Next]**.

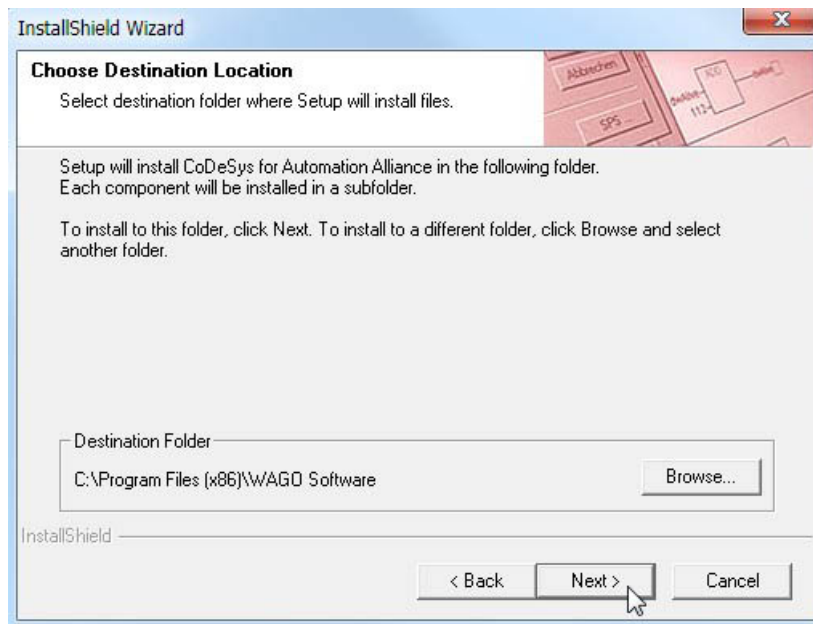


Figure 8: Selecting the target directory

7. Select the components to be installed. Then confirm this selection by clicking **[Next]**.

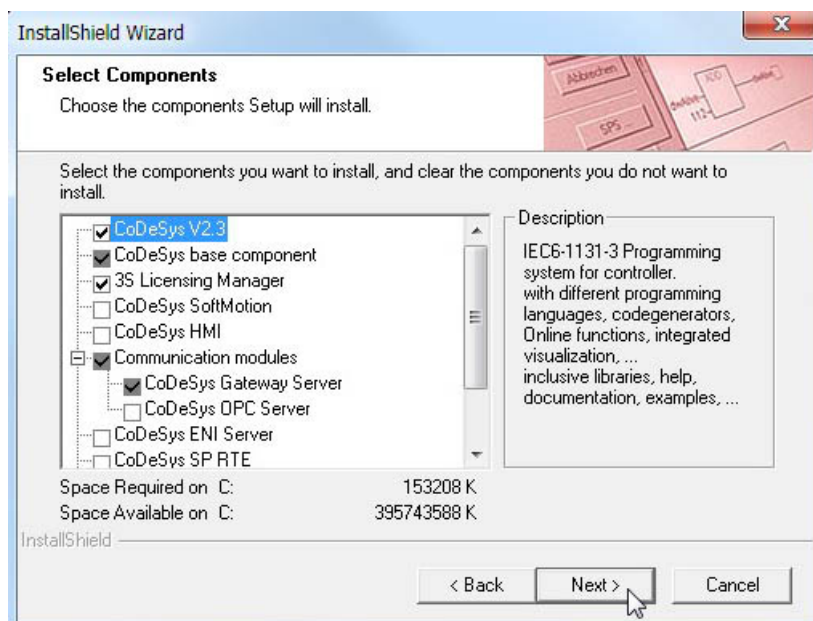


Figure 9: Selection of components

8. Select the program folder. Then confirm this selection by clicking **[Next]**.

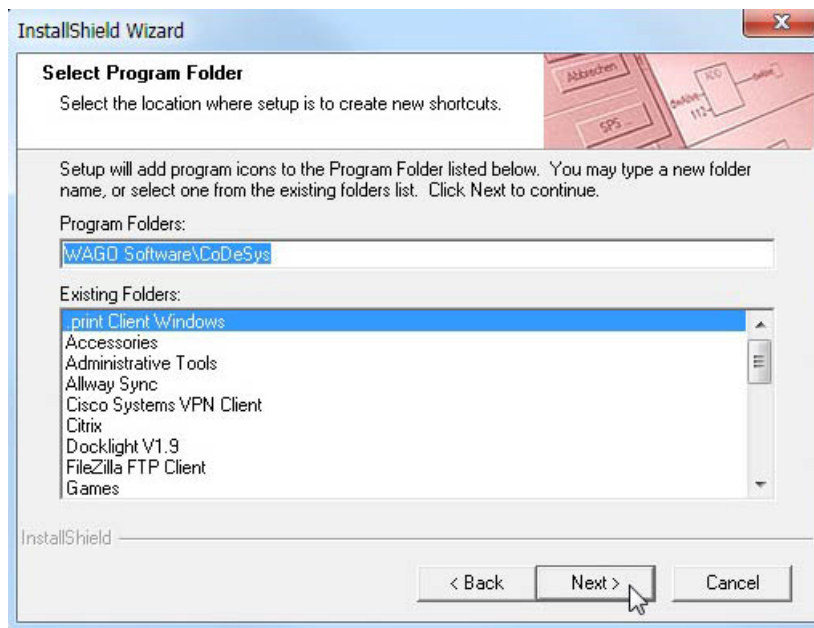


Figure 10: Selection of program folder

9. Check the settings you have defined. Then confirm this selection by clicking **[Next]**.

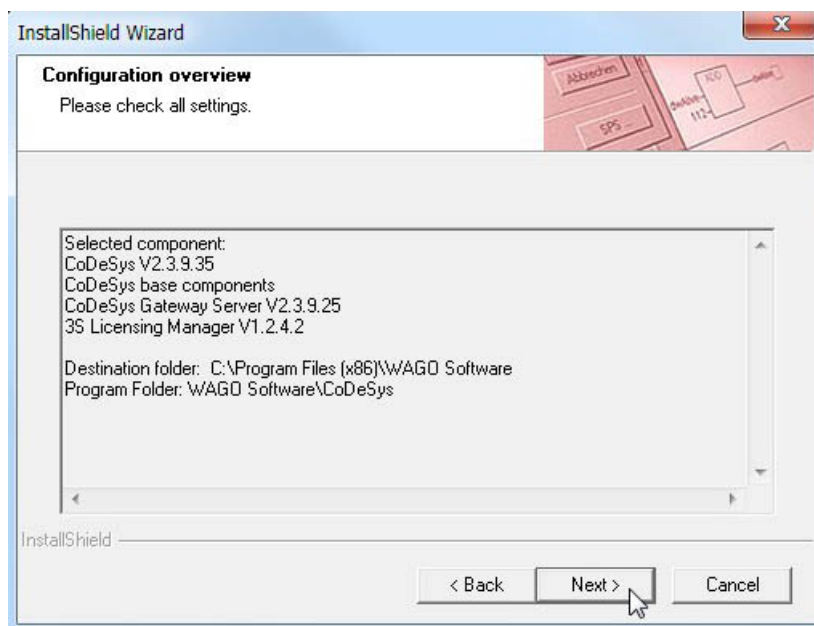


Figure 11: Overview of configuration

This opens the pdf document "EA_ConfigurationReleaseInfo.pdf", which has information about the new version of the WAGO-I/O Configurator dialog.

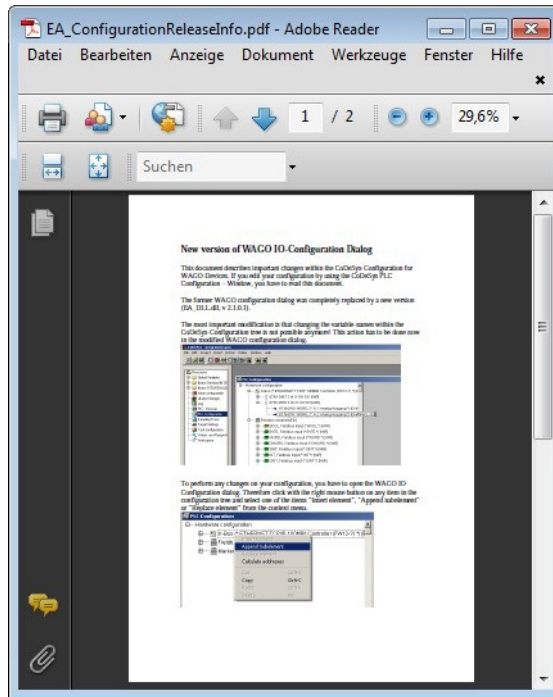


Figure 12: Pdf document "EA_ConfigurationReleaseInfo.pdf"

10. Click [**Complete**] to end setup.

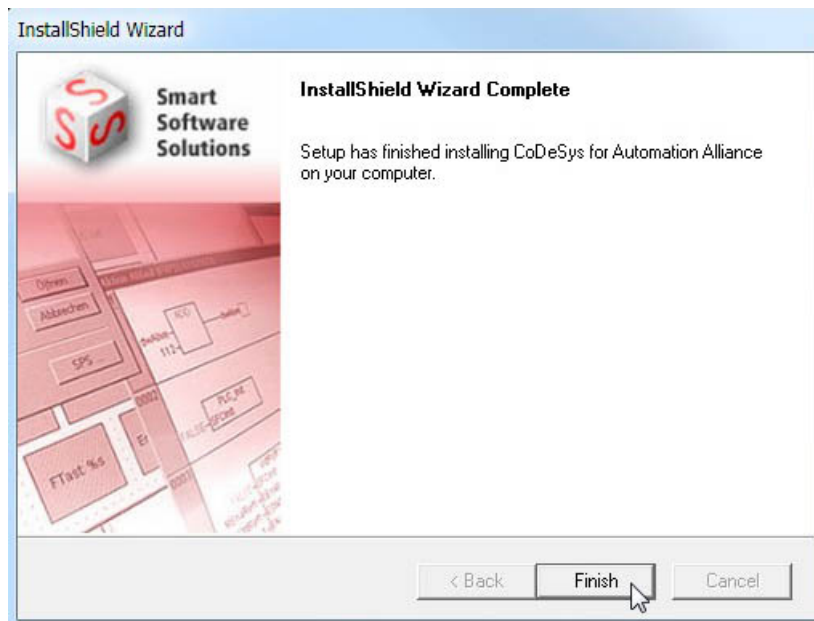


Figure 13: Completing the setup

3.2.3 Starting Up the ETHERNET Network

This section provides you with a step-by-step sample procedure for starting up the ETHERNET network.

You must perform three steps to start up the network:

- **Determine the IP address of your PC and change it if required**
- **Assign an IP address to the fieldbus controller**
- **Test the ETHERNET link**

The description of these individual steps can be found in the corresponding following sections.

Note



Sample description!

This description is just an example. It only describes the procedure for local startup of a single fieldbus controller using a non-networked computer running Windows.

3.2.3.1 Determine the IP address of your PC and change it if required

The explanation below describes how to determine the IP address currently set for your PC and how to change it if required.

1. Connect the ETHERNET port of your PC to one of the two ETHERNET ports of the fieldbus controller using the patch cable provided with the system in order to set up an ETHERNET link.
2. The IP address for your computer can be determined in Windows at: **Start > Control panel > Network and release center > Change adapter settings > LAN connection.** Right click on the option **PROPERTIES** for the network you are currently using (LAN connection X).
3. Mark the item "Internet Protocol Version 4 (TCP/IPv4)" in the popup list and confirm by clicking [**PROPERTIES**].

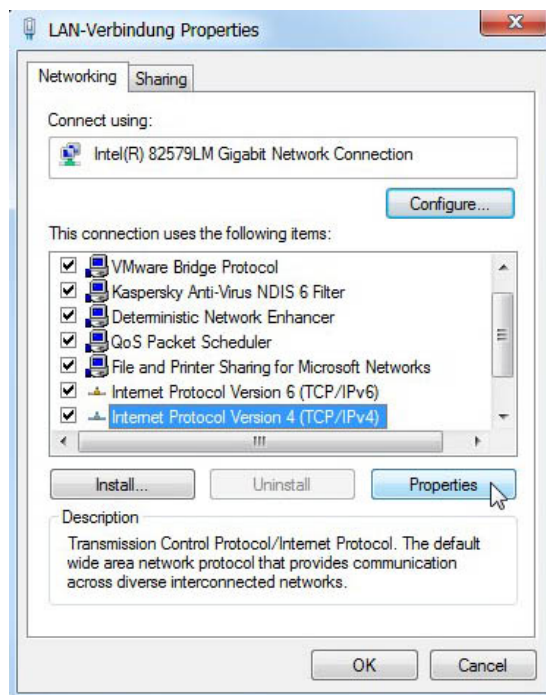


Figure 14: Properties of your LAN connection

4. Check whether the current IP address is set within the address range **192.168.1.1** to **192.168.1.254**. If not, change it accordingly.

Note



Do not use IP address 192.168.1.100!

Do not use the IP address **192.168.1.100**, however, as this network address is reserved for the fieldbus controller in the present example.

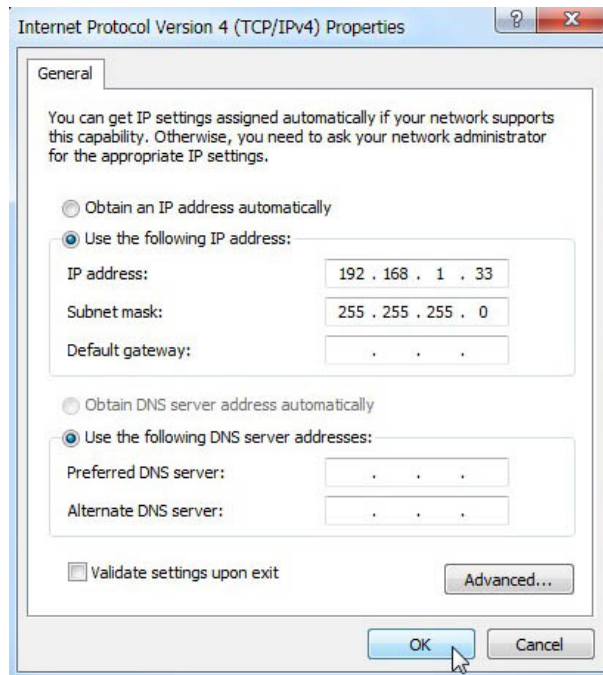


Figure 15: Properties of the TCP/IP Internet protocol

3.2.3.2 Assigning an IP address to the fieldbus controller



Note

The IP address must be unique in the network!

To ensure error-free network communication, the IP addresses assigned to the PC and to the fieldbus controller must be unique within the network!

In the event of an error, the following error message "IP address configuration error" (error code 6 - error argument 6) is indicated by the I/O LED at the fieldbus controller restart.

An IP address for the fieldbus controller can be assigned using the following setting possibilities:

- Address selection switch (DIP switch),
- WebBasedManagement (WBM),
- DHCP,
- BootP or
- WAGO ETHERNET settings.

The option for using the address selection switch is detailed in this QuickStart Guide. The other options are not described here, as these are explained completely in the manual for the "750-881 ETHERNET Programmable Fieldbus Controller".

Use the address selection switch to set the host ID (the last byte of the IP address), which is binary encoded in the range between 1 and 254.

The network address consists of the first three bytes of the IP address; the factory default setting for this is **192.168.1**. You can change this network address in the WebBasedManagement system under menu item **TCP/IP** in the dialog input field "**Switch IP address.**"

Table 6: IP Address Structure

Network Address			Host ID
192	.	168	. 100

The IP address statically set previously by WebBasedManagement will be overwritten with the address assigned by the address selection switch.

Note



Values 0 and 255 are predefined!

Set the values 0 or 255 using the address selection switch. This activates the IP modes provided. When a value of 0 is set, the IP configurations executed by WebBasedManagement will apply. Setting a value of 255 will activate address allocation by DHCP.

- To assign a value (except for 0/255) for the host ID using the address selection switch, you must first convert the host ID value to a binary numeral.
In the configuration example shown here, the host ID 100 is used for the fieldbus controller. The decimal value of "100" converts to a binary code of "01100100."

Table 7: Binary Coding of Decimal Number 100

MSB								LSB
0	1	1	0	0	1	0	0	0

- Set the bits in order using the 8 DIP switches. Begin with DIP switch 1 to set Bit 0 (LSB) and continue up to DIP switch 8 for Bit 7 (MSB).

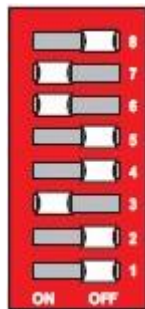


Figure 16: Address selection switch with set IP address (192.168.1.100)

- Implement a restart of the fieldbus controller if you have adjusted the address selection switch, so that the changed IP configuration will be adopted.
To do this, open the cover for the service interface and press the mode switch down (e.g., using an actuation tool, Item No.: 210-720).

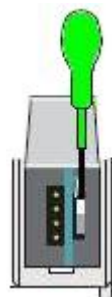


Figure 17: Restarting the fieldbus controller via hardware reset

3.2.3.3 Testing the ETHERNET link

1. Open the command line interpreter by entering "cmd" at **Start > Execute...** in Windows.

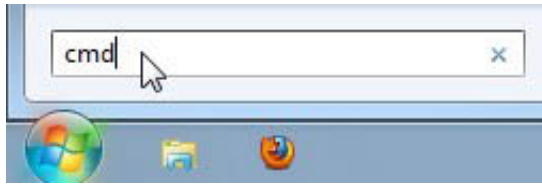


Figure 18: Dialog box "Execute"

2. Enter the command "ping 192.168.1.100."



Figure 19: Command line interpreter with ping command

When the ETHERNET link is error free, you will receive the following ping statistics:

"Packets: Sent=4, Received=4, Lost=0 (0% loss)."

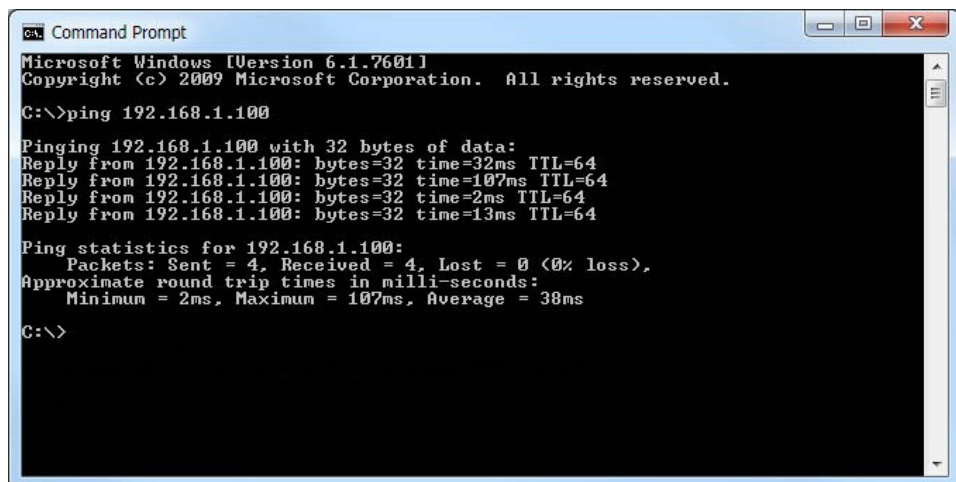


Figure 20: Command line interpreter with ping statistics

3.2.4 Installing the USB Driver

The "WAGO USB Service Cable" (Item No.: 750-923) provides a simple option for communication with WAGO software tools (WAGO-ETHERNET-Settings, WAGO-I/O-CHECK 3, etc.) and with the IEC-61131 programming topology (WAGO-O/O-PRO).

Proceed as follows to install the USB driver:

1. Place the "AUTOMATION Tools and Docs" CD into your CD drive.
2. In the "Software" folder, open the archive file *750-923.zip* and execute the file *Setup.exe* in the "USB driver 759-923" folder.

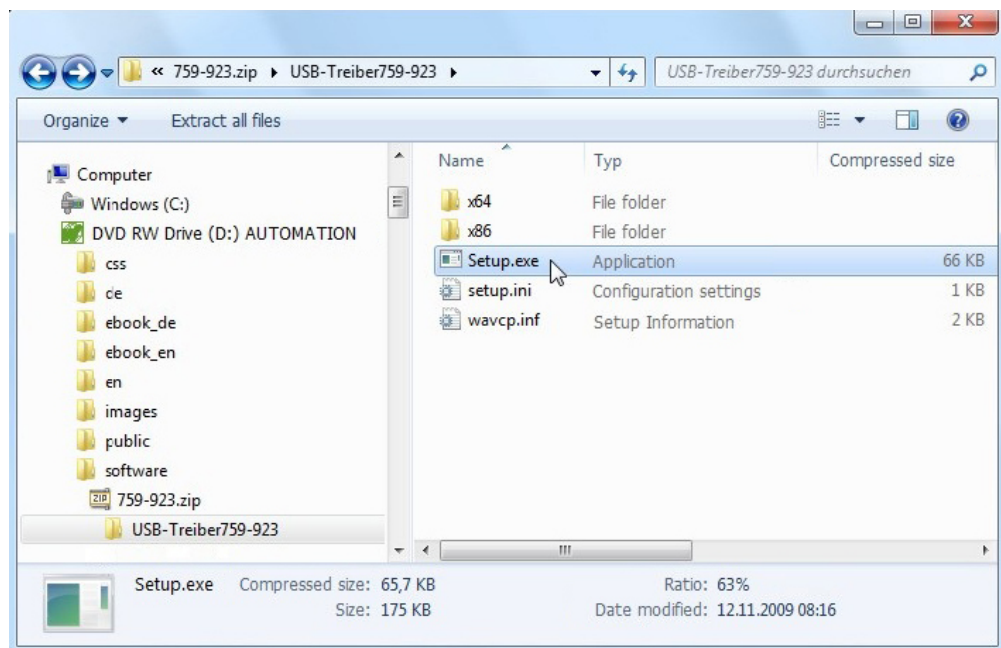


Figure 21: Installing the USB Driver

3. Select the installation directory and start the setup.

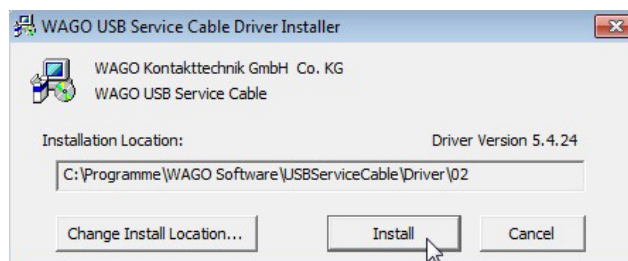


Figure 22: Selecting the target directory

4. Click on the **[OK]** button to finish the installation.

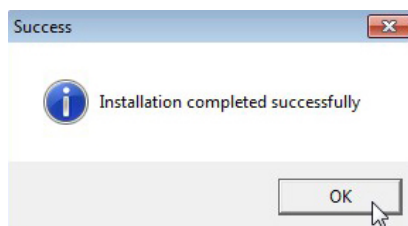


Figure 23: Finishing the Installation

5. Plug the USB cable into an available USB port on your PC.

4 Application example: Starterkit881_App1_FUP.pro

The WAGO-I/O-PRO project "*Starterkit881_App1_FUP.pro*" clearly outlines how the digital inputs and outputs are activated and processed.

Setting and resetting of the digital outputs and indication of the signal statuses of the digital inputs is shown by visualization.

The WAGO-I/O-PRO project "*Starterkit881_App1_FUP.pro*" has been written in the programming languages FUB, KOP, and ST. All three versions of the program have the same functions and differ only in the programming language that is used.

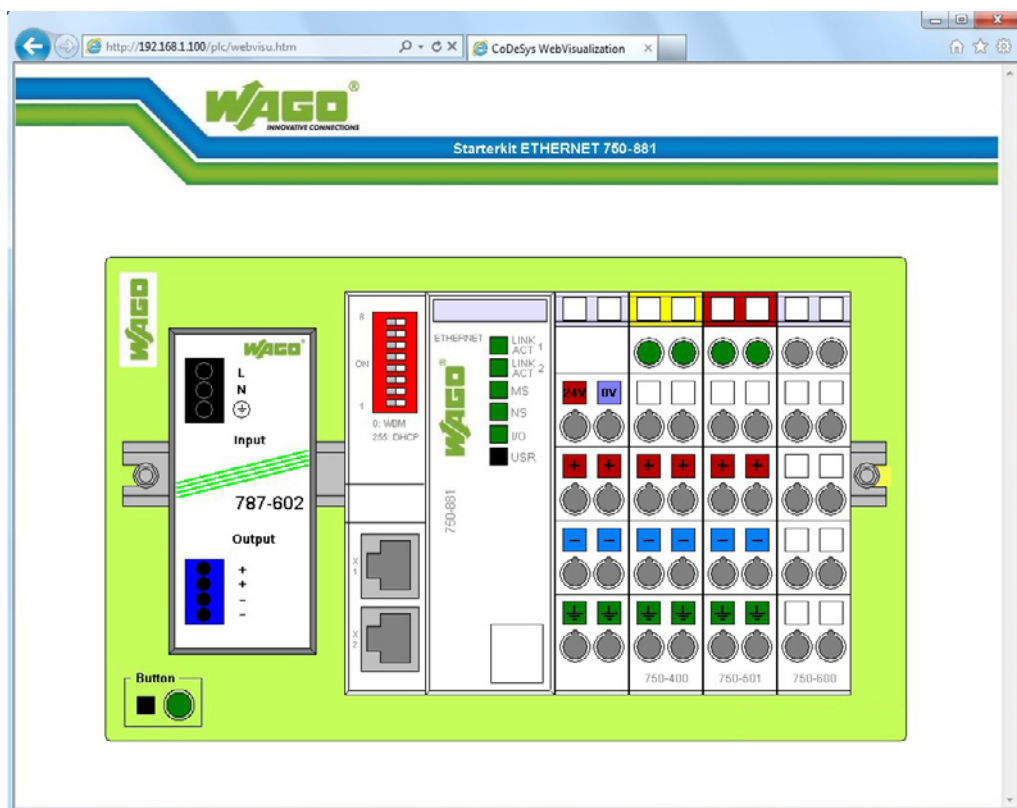


Figure 24: Web visualization

Note



Check steps that are already completed!

The steps described below require that all the steps cited previously have been successfully completed.

1. Documentation and sample projects are downloaded,
2. WAGO-I/O-PRO software and USB cable driver are installed,
3. Link between fieldbus controller and PC is established.

4.1 Open project

1. Start WAGO-I/O-PRO:
Start > All Programs > WAGO-Software > CoDeSys > CoDeSys V2.3 > CoDeSys V2.3.
2. In the WAGO-I/O-PRO software, select **File > Open ...** in the menu and open the CoDeSys project "*Starterkit881_App1_FUB.pro*".

4.2 Configure hardware

1. Select **PLC configuration** in the tree structure in the **Resources** tab in order to access hardware configuration.
2. Double-click "PLC configuration" to open the dialog box.
3. In this dialog, open the complete tree structure for the **K-Bus¹** entry. The 750-400 and 750-501 I/O modules will appear. The associated channels are listed below the I/O module entries.

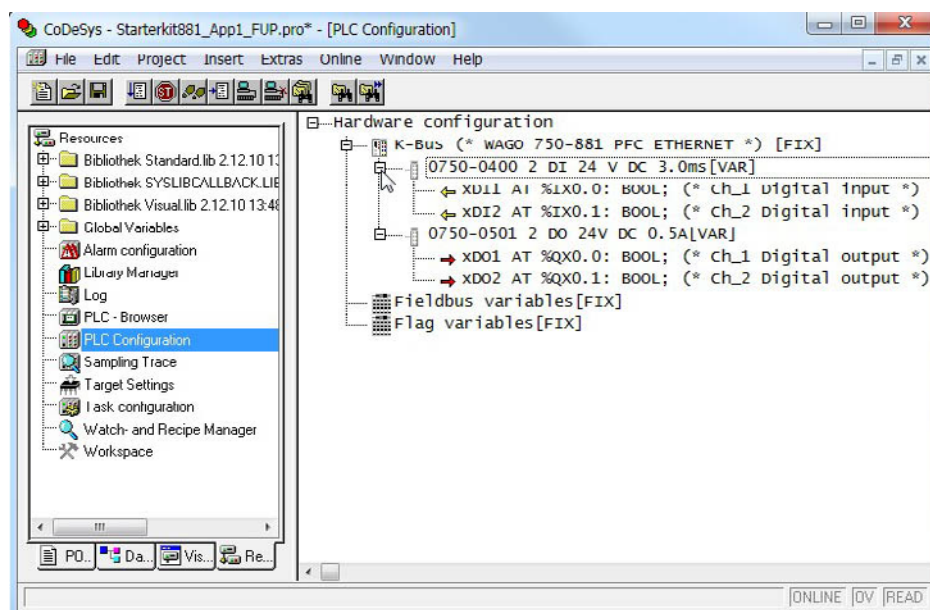


Figure 25: PLC configuration

¹ *K-Bus = internal data bus

4. Select **K-Bus** in the tree structure. Then right-click to open the context menu and select **Add sub-module**. You can change the current PLC configuration.

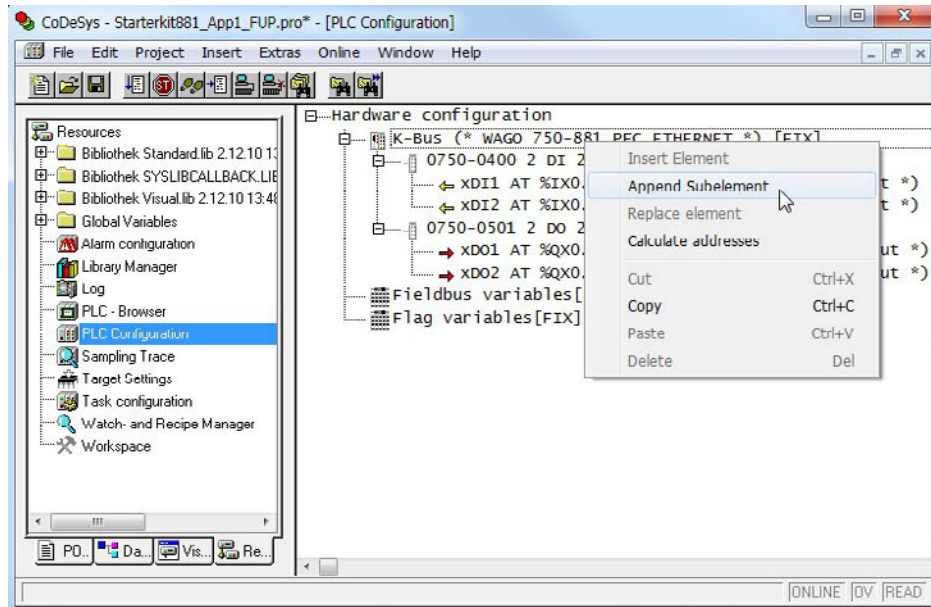


Figure 26: PLC Configuration – Add sub-module

5. Select the **Inputs/Outputs** tab in the "I/O Configurator" dialog. Then click on the [+] button in this tab.

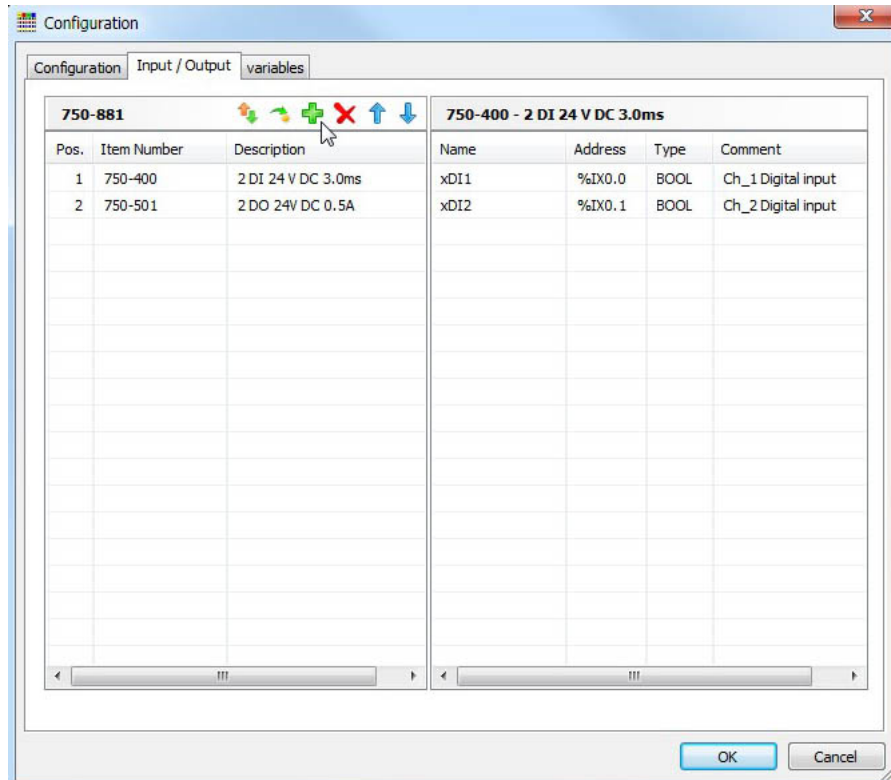


Figure 27: PLC configuration – "Inputs/Outputs" tab

6. In the left window, select the I/O modules based on your hardware setup. Click on the [>>] button to add the selected modules.

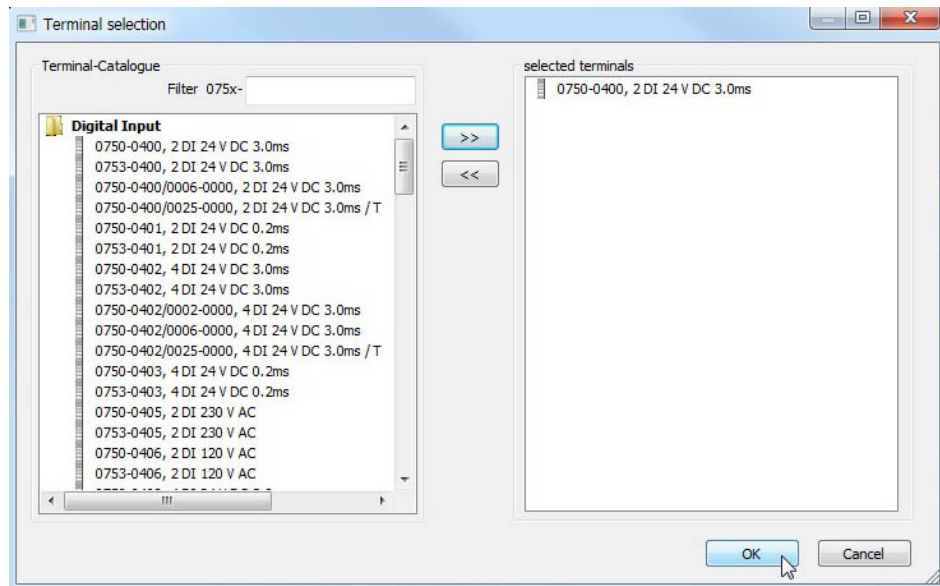


Figure 28: PLC configuration – Select I/O modules

Note



The end module is not specified!

Please note that the 750-600 end module is not specified when configuring the PLC.

7. Assign corresponding variables to the digital input/output modules in the right window of the I/O Configurator. This symbolic name will then be available as a global variable in the overall WAGO-I/O-PRO project. The variables are structured as follows: symbolic name, IEC address, data type, associated comments.

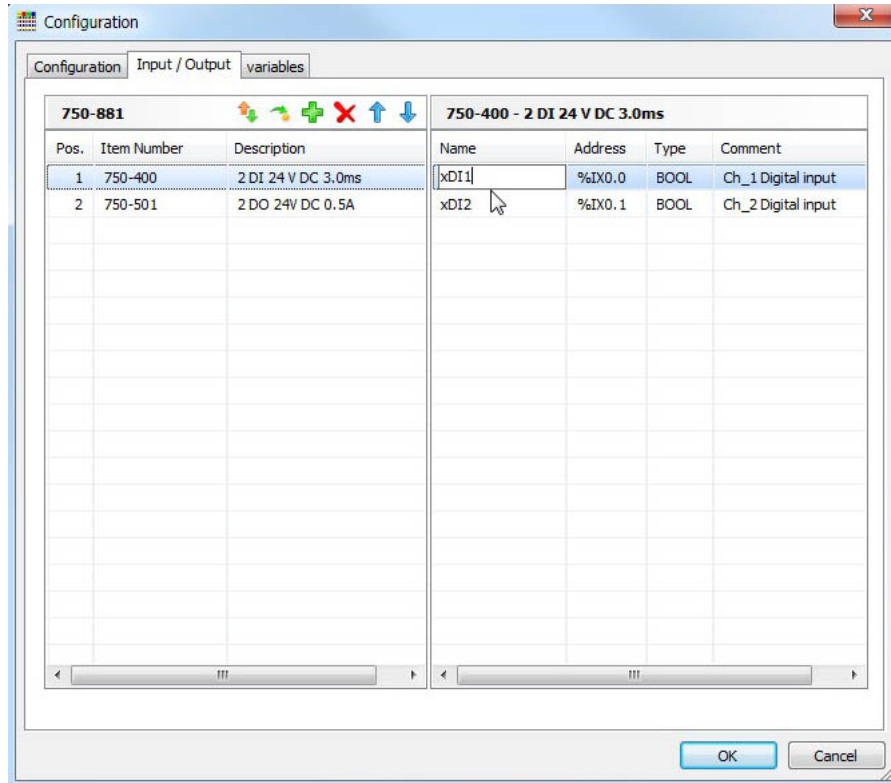


Figure 29: PLC configuration – Assign variables

4.3 Main Program "PLC_PRG"

Select the object **PLC_PRG** in the **Modules** tab in order to open the main program.

You will see the variable declaration in the upper part of the dialog box. The individual FUB networks are displayed in the lower part. You can activate the digital outputs (Network 1 and Network 2) and call up the subprograms "UserLED" and "Blink" (Network 3 and Network 4) in the main program.

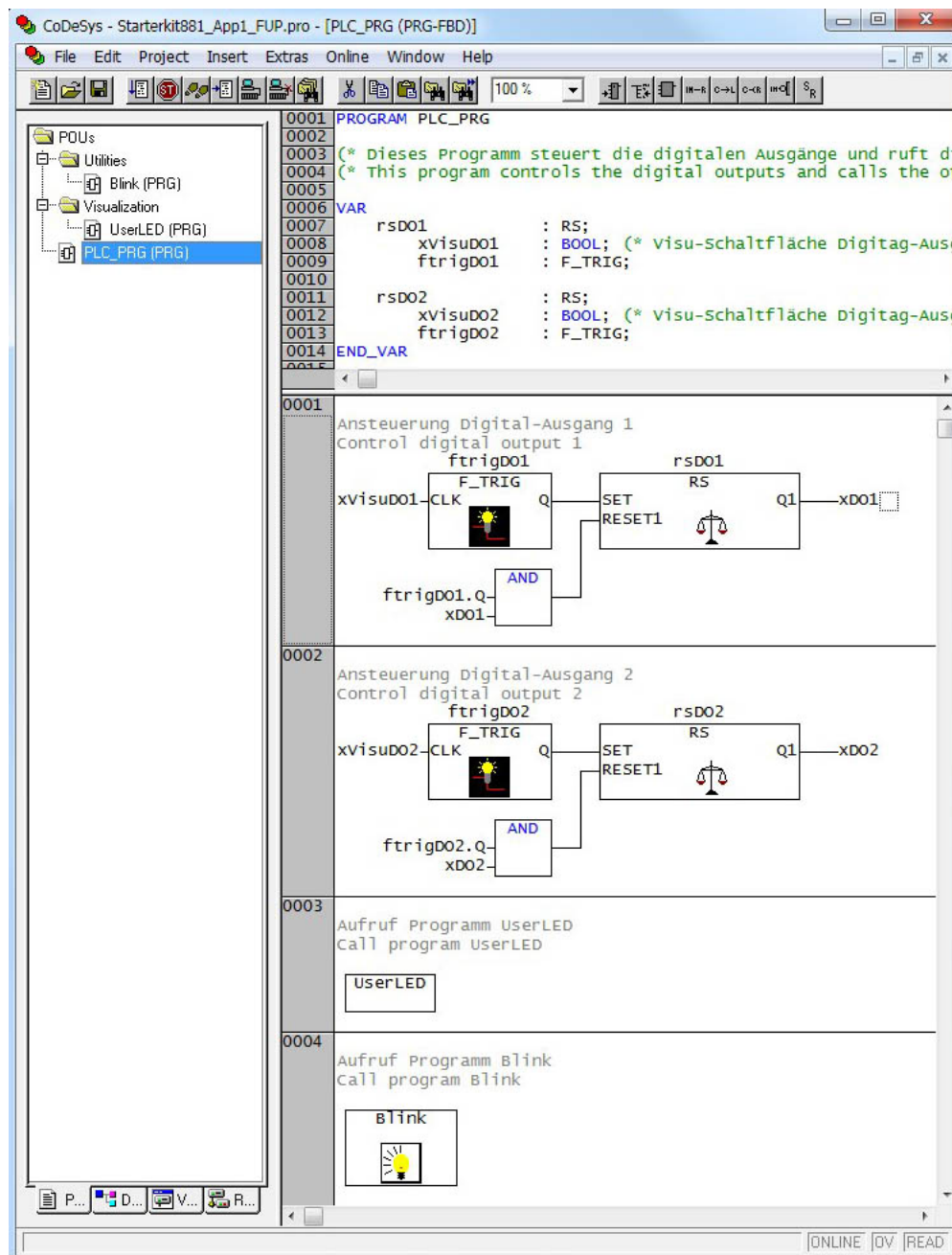


Figure 30: "PLC_PRG" program

You can activate the first digital output using a bistable function block "rsDO1" applying RS logic (reset = dominant).

The "SET" input of the "rsDO1" function block is linked to a flank-detecting function block "ftrigDO1", which detects the falling flank for "xVisuDO1".

The "RESET1" input is linked via a logical AND gate that evaluates the signal for flank detection "ftrigDO1.Q" and the status of the first digital output "xDO1."

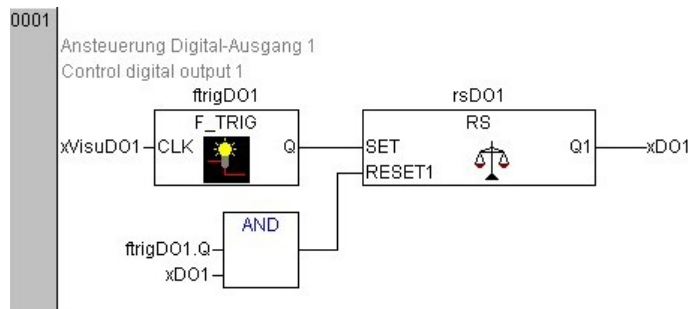


Figure 31: Network 1 (PLC-PRG)

This linking of flank detection, AND gate, and bistable flip-flop yields the following function:

- If the digital output "xDO1" has not been set, then clicking the corresponding visualization button [xVisuDO1] will set the digital output "xDO1."
- If the digital output "xDO1" has already been set, then clicking the visualization button [xVisuDO1] will result in the digital output "xDO1" being reset.

4.4 Subprogram "Blink"

The "Blink" subprogram is located in the **Utilities** folder.

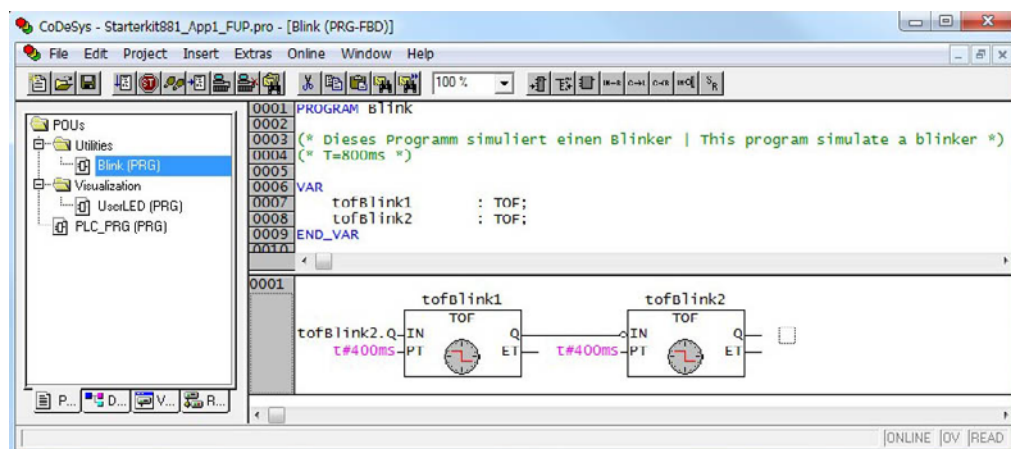


Figure 32: "Blink" program

A pulsing signal is generated in the "Blink" subprogram. This blink signal "tofBlink2.Q" is required for the flashing frame around the buttons in the display. The pulse period for this signal is 800 ms, which corresponds to a frequency of 1.25 Hz.

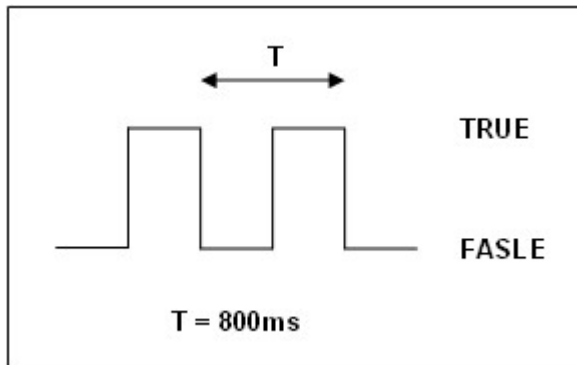


Figure 33: Blink signal

The clock-pulse generator is implemented using two OFF delays connected in series, "tofBlink1" and "tofBlink2." This is achieved by linking the output "Q" of "tofBlink1" to the negated input "IN" of "tofBlink2." In addition, the output "Q" of "tofBlink2" is also linked back to the input "IN" of "tofBlink1."

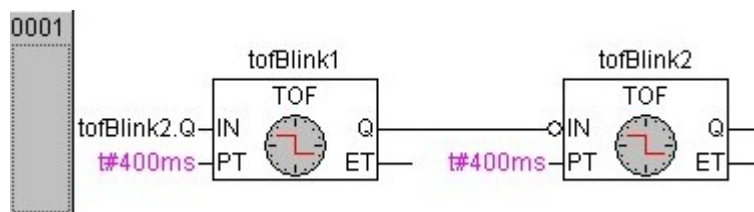


Figure 34: Network 1 (Blink)

The signal status sequence for the two OFF delays "tofBlink1.Q" and "tofBlink2.Q" is illustrated as follows:

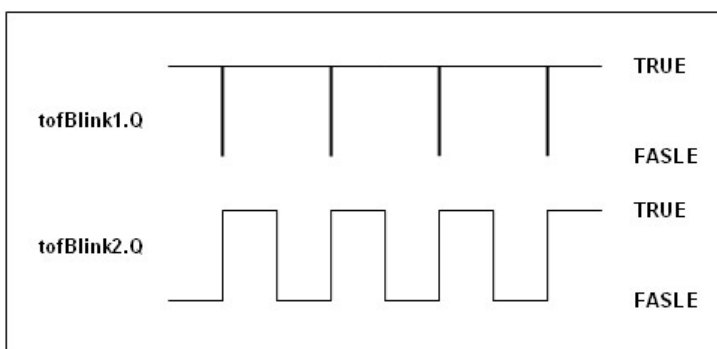


Figure 35: Output signals for OFF delays

4.5 Subprogram "UserLED"

Note



Required library "Visual.lib"!

You must have the "Visual.lib" library to activate the USR-LED using the functions described below.

This library is located in the directory:

WAGO Software > CoDeSys V2.3 > Targets > WAGO > Libraries > 32_Bit

The "UserLED" subprogram is located in the **Visualization** folder. The "UserLED" subprogram activates the USR-LED fieldbus controller. Changing the USR-LED colors is carried out using a button in the visualization. This LED can be displayed in green, red and orange.

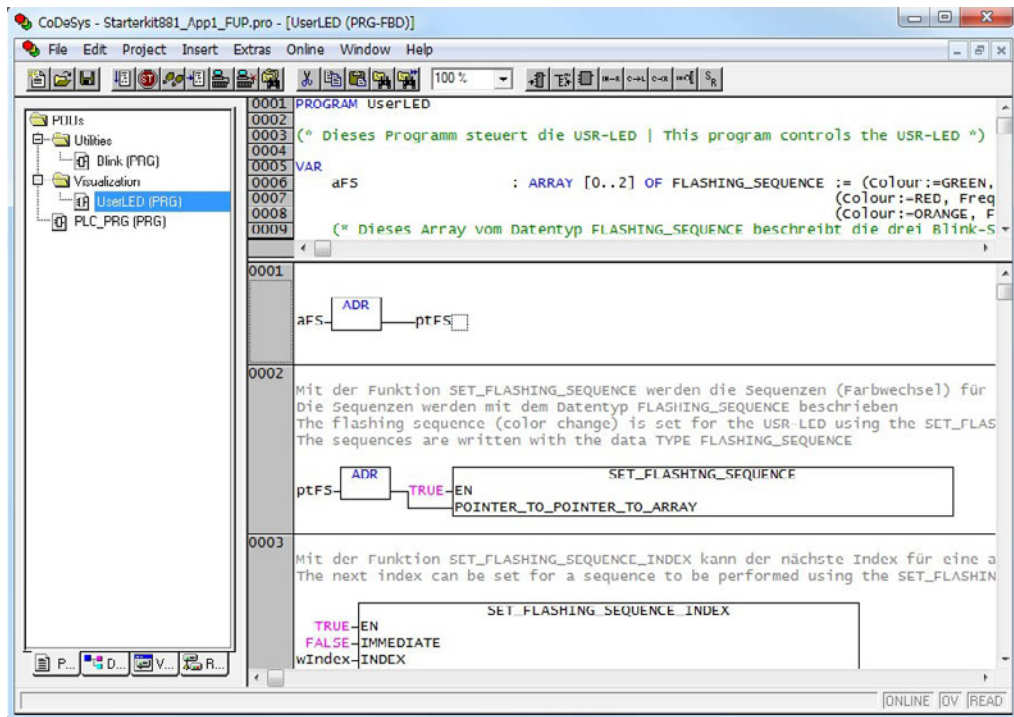


Figure 36: "UserLED" program

Information



Further information for activation of the USR-LED

Activation of the USR-LED is only explained with an example in this section. For more detailed information about USR-LED signaling and about using the "Visual.lib", the corresponding library description will be of assistance.

The help file for this description is located in the WAGO-I/O-PRO directory:

WAGO Software > CoDeSys V2.3 > Targets > WAGO > Help > German > CAA-WAGO_Visual.chm.

In the first network, the address operator "ADR" supplies the address for the variable "aFS" and stores this in the pointer variable "ptFS." The individual blinking sequences for the USR-LED (LED colors) are defined by the array variables "aFS."

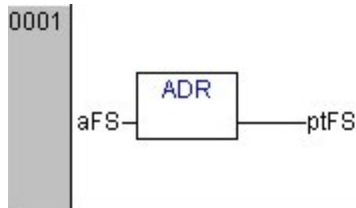


Figure 37: Network 1 (USR-LED)

In the second network, the flashing sequences for the USR-LED are set by the function "SET_FLASHING_SEQUENCE." The input parameter "EN" (Enable) enables execution of this function. The input parameter "POINTER_TO_POINTER_TO_ARRAY" transfers the address of the pointer variables "ptFS" and by this means the individual flashing sequences as well.

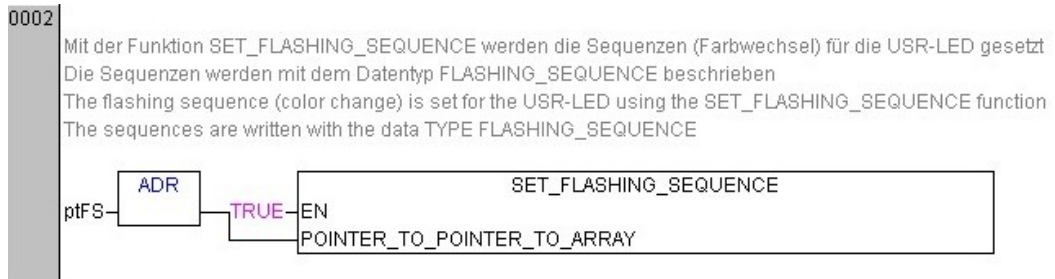


Figure 38: Network 2 (USR-LED)

In the third network, the next index is set for a flashing sequence to be executed using the function "SET_FLASHING_SEQUENCE_INDEX." The input parameter "IMMEDIATE" determines when the new flashing sequence should start:

- **FALSE:** the new sequence is initiated after the end of the current sequence
- **TRUE:** the new sequence begins immediately. The parameter "INDEX" defines the index for the new flashing sequence.

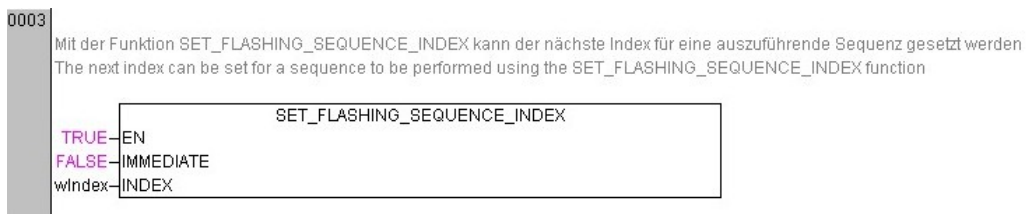


Figure 39: Network 3 (USR-LED)

The function "START_FLASHING_SEQUENCE" starts the sequences set previously using the function "SET_FLASHING_SEQUENCE." Index 0 (USR-LED = green) is executed first, as the variable "wIndex" was initialized with this value in the variable declaration.

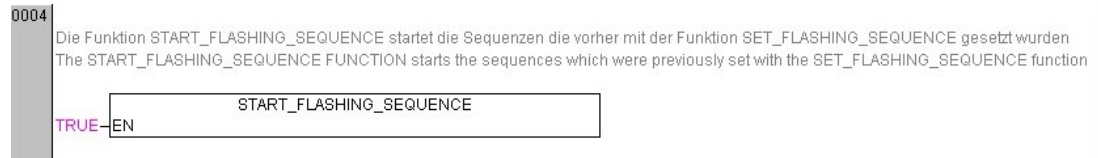


Figure 40: Network 4 (USR-LED)

Clicking the visualization button [USR-LED] increases the value of "wIndex" by "1" (from 0 to a maximum of 3).

To accomplish this, the variable "xVisuUserLed" for the visualization button [USR-LED] is initially processed via flank detection "ftrigVisuUserLed" and then evaluated as follows via selection (SEL):

- If "ftrigVisuUserLed.Q" = **TRUE**, the selection result is `1`.
- If "ftrigVisuUserLed.Q" = **FALSE**, the selection result is `0`.

This result is subsequently added to the value of "wIndex" (ADD), then compared to the value "3" (GT = greater than) and finally evaluated as follows via selection:

- If the addition result is greater than "3", then "wIndex" is assigned the value "0".
- If the addition result is lower than or equal to "3", then "wIndex" is assigned the result of this addition.

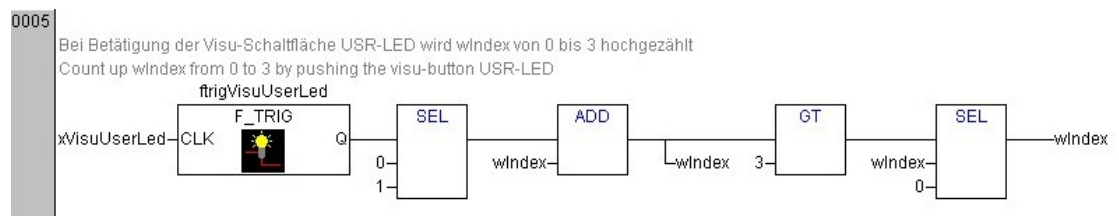


Figure 41: Network 5 (USR-LED)

In the sixth Network, the visualization button **[USR-LED]** is assigned a color as a function of "wIndex." This ensures that the **[USR-LED]** button has the same color in the display as the physical USR-LED for the fieldbus controller. This value is assigned via a multiplexer (MUX).

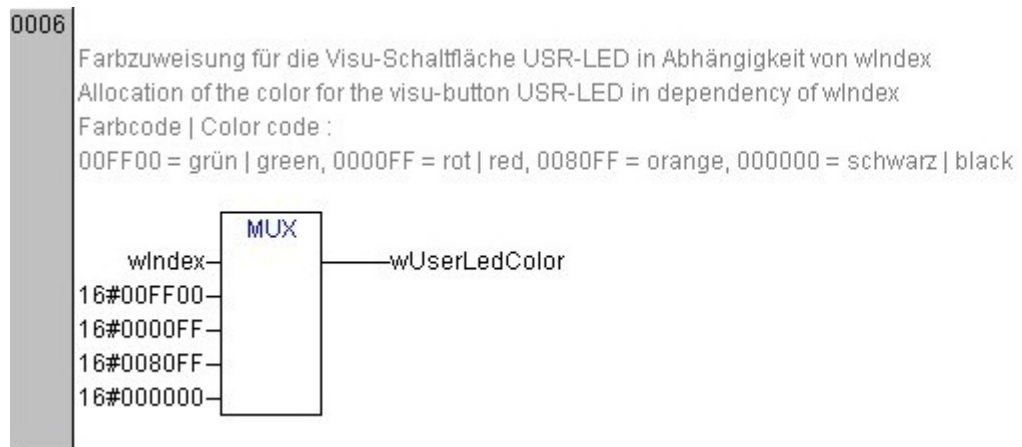


Figure 42: Network 6 (USR-LED)

Table 8: Button Color Allocation **[USR-LED]**

"wIndex"	"wIndex"	"wIndex"
0	0	0
1	1	1
2	2	2
3	3	3

4.6 Commissioning

The following section provides a step-by-step description of how to load the WAGO-I/O-PRO project "Starterkit881_App1_FUP.pro" to your fieldbus controller, how to start program execution, and how to create a boot project.

4.6.1 Configuring a Communication Driver for an ETHERNET Link

1. To configure the communication driver for the ETHERNET link, click the option **Communication parameters** in the **Online** menu.

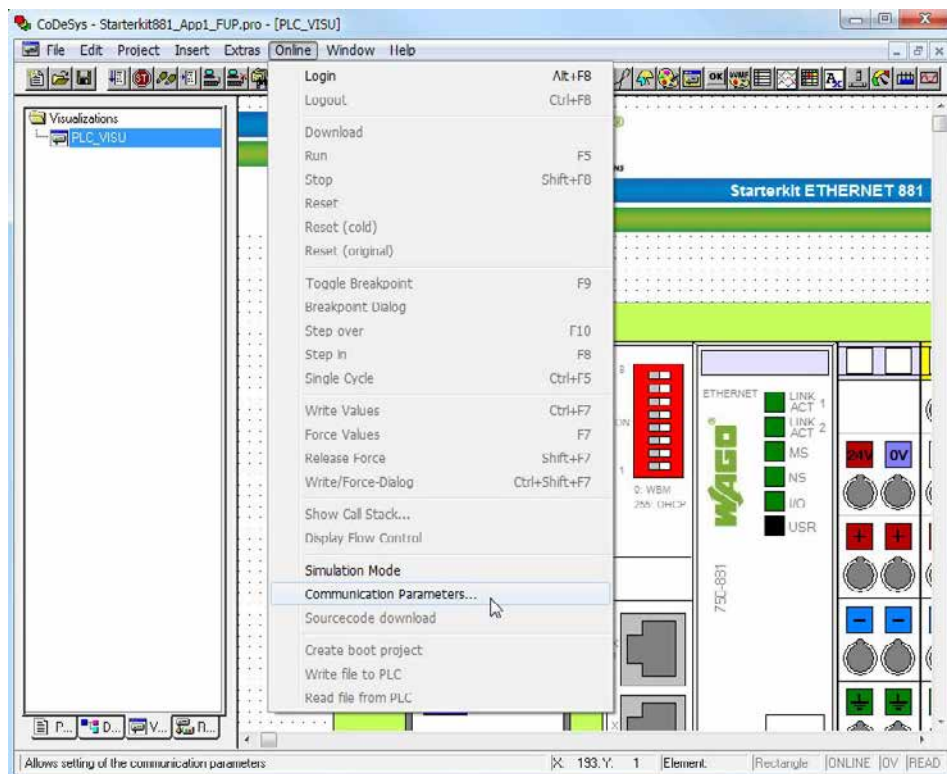


Figure 43: Communication parameters

2. Then click the **[NEW...]** button in the "Communication parameters" dialog box to create a new communication channel.

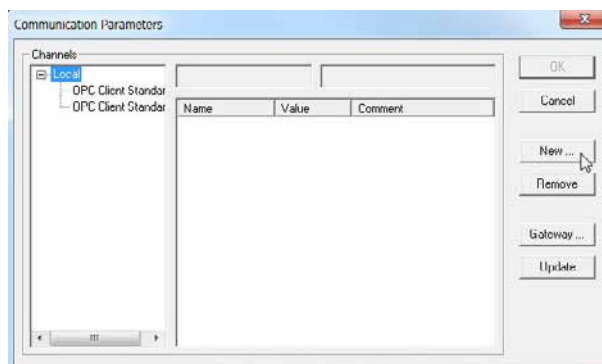


Figure 44: Communication Parameters – Create communication channel

3. Enter a name for the ETHERNET link in the **Name** field.
4. Select the "Tcp/Ip 3S Tcp/Ip driver" and confirm your input by clicking **[OK]**.

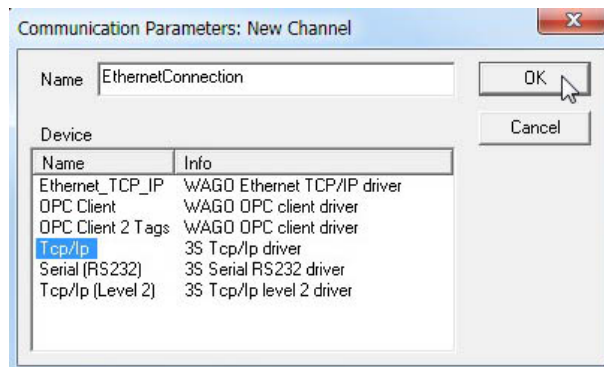


Figure 45: New communication channel via ETHERNET

5. Double-click on the address field **localhost** in the column "Value" in the "Communication parameters" dialog box.
6. Here, enter the IP address of your fieldbus controller "192.168.1.100" and confirm this using **[Enter]**.
You do not need to make any changes to the other parameters (Port, Motorola byteorder) or to the gateway settings. Confirm your input by clicking **[OK]**.

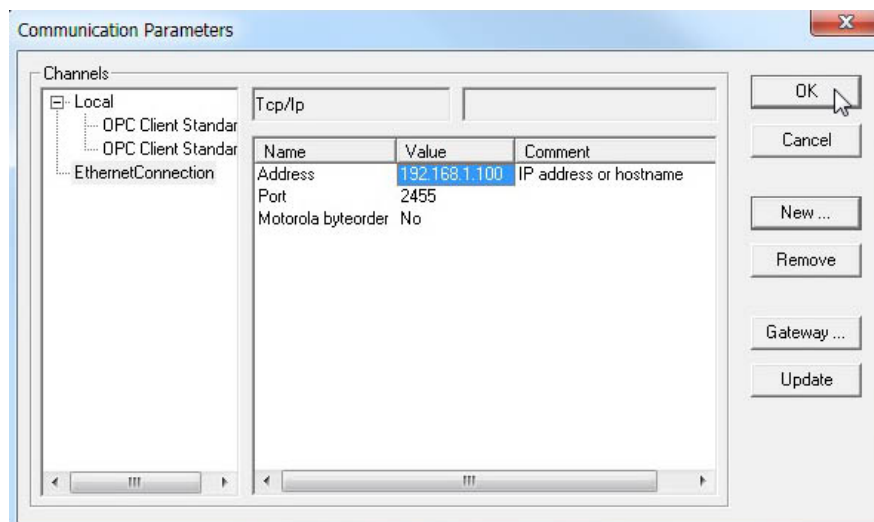


Figure 46: ETHERNET communication parameters

4.7 Configuring a Communication Driver for a USB Link

1. In Windows, you can check which port of your PC has been assigned to the USB cable at:
Start > Control panel > System > Device manager.
2. Add the group **Ports (COM and LPT)** to the tree structure.
The entry **WAGO USB Service Cable (COMX)** is given within this group.
The designation in parentheses indicates the assigned COM port; in this example "COM3".

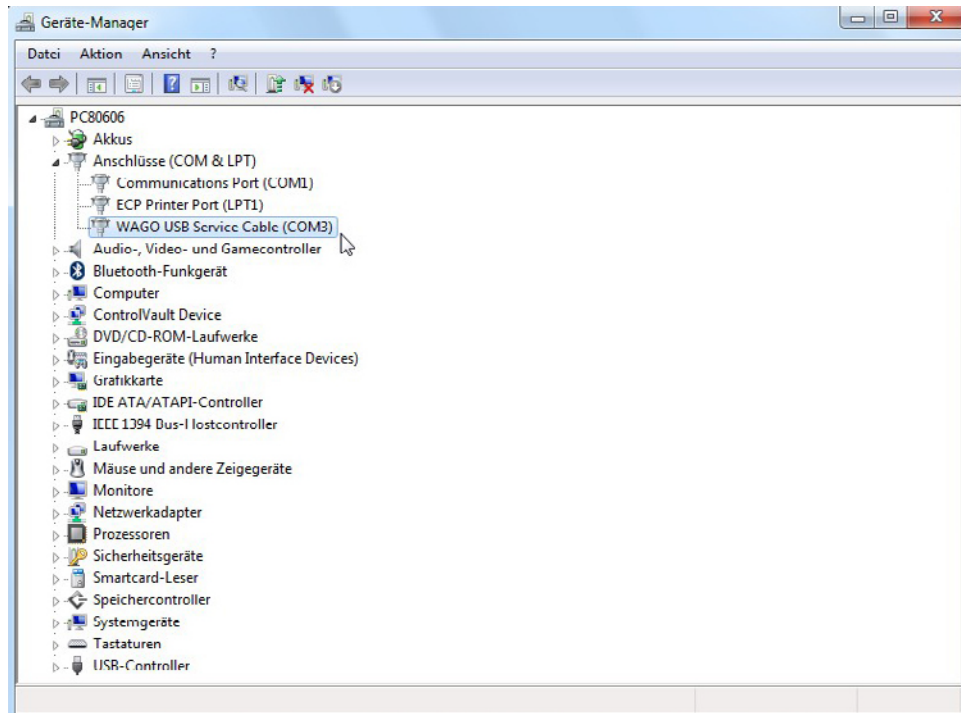


Figure 47: Device manager using Windows

3. To configure the communication parameters for the USB link, click the item **Communication parameters** in the **Online** menu.
4. In the "Communication Parameters" dialog, click the [NEW...] button to create a new communication channel.
5. Enter a name for the USB link in the **Name** field.

6. Select the driver "Serial (RS232) 3S Serial RS232 driver" and confirm your input by clicking [OK].

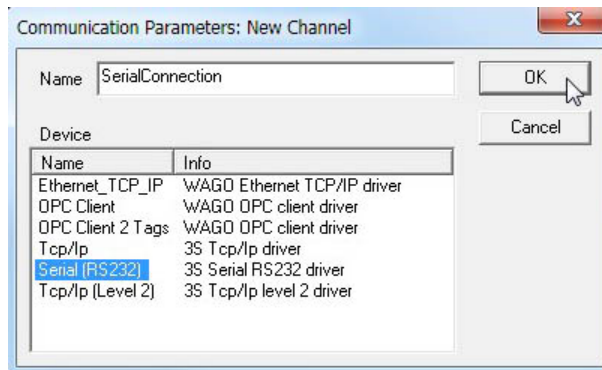


Figure 48: New communication channel via USB

7. Double-click on the address field **COM1** in the column "Value" in the "Communication parameters" dialog box. The field is highlighted in gray.
8. Press the [↓] or [↑] arrow to increase or decrease the COM port number by one. Adjust the COM port, which was previously identified in the Device Manager (in this example "COM3"), accordingly in the Communication Parameters.
You can leave all other parameters (Baudrate, Parity, Stop bits, Motorola byteorder, Flow Control) and the gateway settings as they are. Confirm your input by clicking [OK].

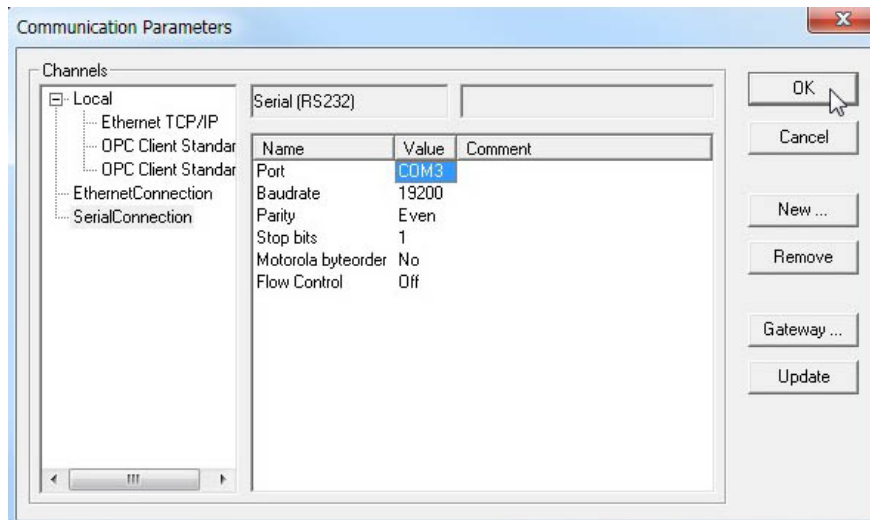


Figure 49: USB communication parameters

4.7.1 Starting the Program

1. Click **Login** in the **Online** menu to load this program to your fieldbus controller.
2. Then start program execution in the fieldbus controller via the menu item **Online > Start**.
3. Now select the menu item **Online > Create bootproject** to create a boot project. The program is now automatically loaded when restarting the fieldbus controller.
4. Switch the Operating mode switch for the fieldbus controller to the top position (RUN) so that the fieldbus controller automatically starts up during a restart.

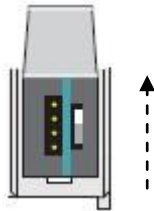


Figure 50: Operating mode switch in RUN position

4.8 Visualization

Two options are available for visualization operation:

- You can directly control the visualization via the *WAGO-I/O-PRO* software after you have loaded and started the program on the fieldbus controller.
- You can also call up and control the process in an Internet browser as a Web visualization.

The function and presentation are identical for both visualization options.

4.8.1 Opening Visualization via the *WAGO-I/O-PRO* Software

1. Select the **Visualizations** tab in the left column and open the **Visualization** folder.
2. In this folder select the object "PLC_VISU."

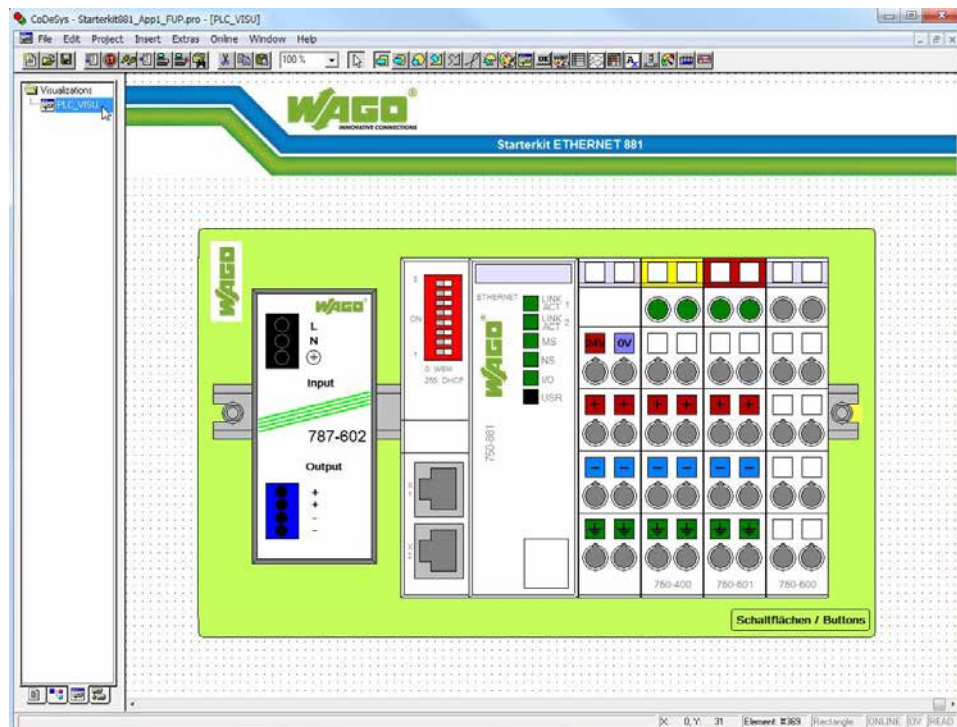


Figure 51: Visualization via *WAGO-I/O-PRO* software

4.8.2 Opening Web Visualization in the Internet Browser

1. Enter the IP address for your fieldbus controller (192.168.1.100) in the address line of your browser. You are then transferred to WebBasedManagement for your fieldbus controller.
2. After this, select the menu item **WebVisu** at the left in the navigation area. You can also call up Web visualization directly via *http://192.168.1.100/plc/webvisu.htm*.

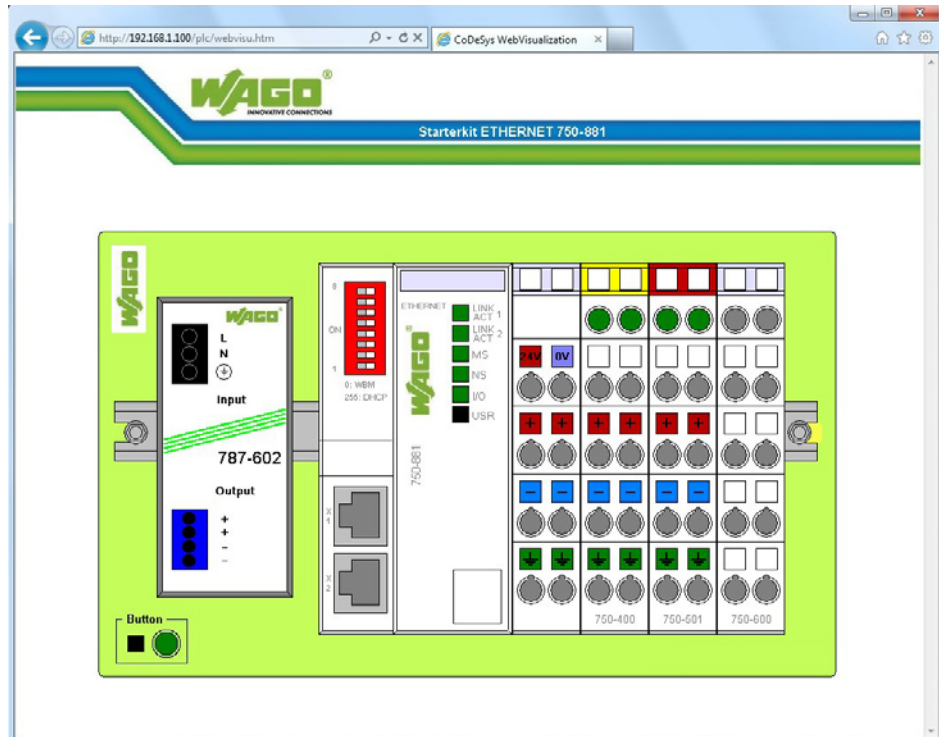


Figure 52: Web visualization with an Internet browser

Note



Java runtime environment must be installed for this!

You must have a Java runtime environment installed on your PC to use Web visualization in your Internet browser.

You can download the Java runtime environment ("JRE") free of charge at:
<http://java.com/de/download/index.jsp>.

4.8.3 Operating the Visualization

Actuatable buttons in the visualization are indicated by a blinking frame. In addition, your mouse pointer changes when you move it over these buttons. Hovering for a short period will call up a yellow box, in which you can read the button function.

You can control the digital output channels via the two buttons displayed for the 750-501 digital output module.

If you use the switch module provided (Item no. 288-863) to control a digital output channel, then the corresponding LED changes its color to the 750-400 digital input module.

The color of the USR-LED can be changed using the **[USR-LED]** button. Each click of the button will change the USR-LED color.

List of Figures

Figure 1: View and 24V/0V connection of the ETHERNET-TCP/IP fieldbus controller	12
Figure 2: Mode selector switch (closed and open damper of the service port).....	14
Figure 3: Download possibilities at the WAGO homepage	15
Figure 4: Contents of the WAGO-I/O-PRO CD (folder view), <i>Setup.exe</i> file.....	16
Figure 5: Selecting the setup language.....	16
Figure 6: Starting the setup	17
Figure 7: License agreement	17
Figure 8: Selecting the target directory	18
Figure 9: Selection of components.....	18
Figure 10: Selection of program folder	19
Figure 11: Overview of configuration.....	19
Figure 12: Pdf document "EA_ConfigurationReleaseInfo.pdf"	20
Figure 13: Completing the setup	20
Figure 14: Properties of your LAN connection.....	22
Figure 15: Properties of the TCP/IP Internet protocol	23
Figure 16: Address selection switch with set IP address (192.168.1.100).....	25
Figure 17: Restarting the fieldbus controller via hardware reset	25
Figure 18: Dialog box "Execute"	26
Figure 19: Command line interpreter with ping command	26
Figure 20: Command line interpreter with ping statistics	26
Figure 21: Installing the USB Driver	27
Figure 22: Selecting the target directory	27
Figure 23: Finishing the Installation	28
Figure 24: Web visualization	29
Figure 25: PLC configuration.....	30
Figure 26: PLC Configuration – Add sub-module.....	31
Figure 27: PLC configuration – "Inputs/Outputs" tab.....	31
Figure 28: PLC configuration – Select I/O modules.....	32
Figure 29: PLC configuration – Assign variables	33
Figure 30: "PLC_PRG" program	34
Figure 31: Network 1 (PLC-PRG)	35
Figure 32: "Blink" program.....	35
Figure 33: Blink signal	36
Figure 34: Network 1 (Blink).....	36
Figure 35: Output signals for OFF delays	36
Figure 36: "UserLED" program	37
Figure 37: Network 1 (USR-LED).....	38
Figure 38: Network 2 (USR-LED).....	38
Figure 39: Network 3 (USR-LED).....	38
Figure 40: Network 4 (USR-LED).....	39
Figure 41: Network 5 (USR-LED).....	39
Figure 42: Network 6 (USR-LED).....	40
Figure 43: Communication parameters	41
Figure 44: Communication Parameters – Create communication channel	41
Figure 45: New communication channel via ETHERNET.....	42
Figure 46: ETHERNET communication parameters	42
Figure 47: Device manager using Windows	43

Figure 48: New communication channel via USB	44
Figure 49: USB communication parameters	44
Figure 50: Operating mode switch in RUN position.....	45
Figure 51: Visualization via WAGO-I/O-PRO software	46
Figure 52: Web visualization with an Internet browser	47

List of Tables

Table 1: Number Notation..... 7
Table 2: Font Conventions 7
Table 3: Fieldbus Node Structure..... 12
Table 4: Legend to the View ETHERNET TCP/IP Fieldbus Controller 13
Table 5: Mode selector switch..... 14
Table 6: IP Address Structure 24
Table 7: Binary Coding of Decimal Number 100 25
Table 8: Button Color Allocation [USR-LED]..... 40

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: +49/5 71/8 87 – 0
Fax: +49/5 71/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>



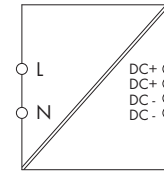
This Page Left Intentionally Blank

WAGO SYSTEM 787-1012
SWITCHED MODE POWER
SUPPLY

Operations & Maintenance Manual
December 2015

Switched-Mode Power Supply

EPSITRON® COMPACT Power



- Primary switch mode power supply unit
- Prepared for class II equipment
- Natural convection cooling when horizontally mounted
- Stage profile, ideal for distribution boards or distribution boxes
- Parallel operation, series connection possible
- Electrically isolated output voltage (SELV) acc. to EN 60950-1/UL 60950

Technical Data

Input:

Nominal input voltage V_i nom	100 ... 240 VAC
Input voltage range	85 ... 264 VAC; 120 ... 373 VDC
Input voltage derating	< 100 VAC: I_a max. 2.0 A
	< 90 VAC: I_a max. 1.8 A
Frequency	44 ... 66 Hz; 0 Hz
Input current I_i	1.4 A at 110 VAC / 0.6 A at 230 VAC
Discharge current	1 mA typ.
Inrush current	< 30 A, NTC
Mains failure hold-up time	> 10 ms at 110 VAC / > 80 ms at 230 VAC

Output:

Nominal output voltage V_o nom	24 VDC (SELV)
Output voltage range	22.8 ... 26.4 VDC adjustable
Output current I_o	2.5 A at 24 VDC
	max. 1.6 A in any mounting position
Factory preset	24 VDC
Adjustment accuracy	2%
Residual ripple	< 100 mVpp at 20 MHz
Current limitation	1.1 x I_o typ.
Overload behavior	Constant current
Operational indication	LED green (V_a)

Efficiency / power losses:

Efficiency	88 % typ.
Power loss P_V	2.2 W (230 VAC/no load), 8.5 W (230 VAC/rated load)
Max. power loss P_V	10.5 W typ. (100 VAC / 24 VDC, 2.5 A)

Fuse protection:

Internal fuse	2 AT
External fuse	Circuit breakers 10 A, 16 A, characteristic: B or C
	An external DC fuse is required for the DC input voltage

Description	Item No.	Pack. Unit
Switched-Mode Power Supply, 24 VDC / 2.5A	787-1012	1

Technical Data

Environmental requirements:

Ambient operating temperature	-25 °C ... +60 °C (UL: -25 °C ... +55 °C); Device start at -40 °C (type-tested)
Storage temperature	-25 °C ... +80 °C
Rel. humidity	5 % ... 96 % (no condensation)
Derating	-3 % / K (> 45 °C)
Degree of pollution	2 (acc. to EN 50178)
Climatic category	3K3 (acc. to EN 60721)

Safety and protection:

Enclosure	Plastic, light gray, Flammability class V0 acc. to UL94
Test voltage pri. - sec.	4.2 kV DC
Protection class	Prepared for class II equipment
Degree of protection	IP20 (acc. to EN 60529)
Overvoltage protection	< 40 VDC (in the event of a fault)
Short circuit protection	yes
No-load proof	yes
Feedback voltage	max. 30 VDC
Parallel operation	yes
Series connection	yes
MTBF	> 500000 h

Connection and type of mounting:

Wire connection	Input/Output: WAGO 740 Series
Cross sections	Input/Output: 0.08 ... 2.5 mm ² / AWG 28 ... 12
Strip lengths	Input/Output: 6 ... 7 mm / 0.24 ... 0.28 in
Type of mounting	DIN-rail mount (EN 60715)

Dimensions and weight:

Dimensions (mm) W x H x L	72 x 89 x 59
	Length: 55 mm, from upper edge of DIN 35 rail
Weight	255 g

Standards and approvals:

Standards/Specifications	EN 60950, EN 61204-3, UL 60950, UL 508, GL
--------------------------	---

This Page Left Intentionally Blank

WAGO SYSTEM 750-513
2-CHANNEL RELAY OUTPUT
MODULE

Operations & Maintenance Manual
December 2015

2-Channel Relay Output Module 230 V AC, 30 V DC

Isolated outputs; 2 make contacts

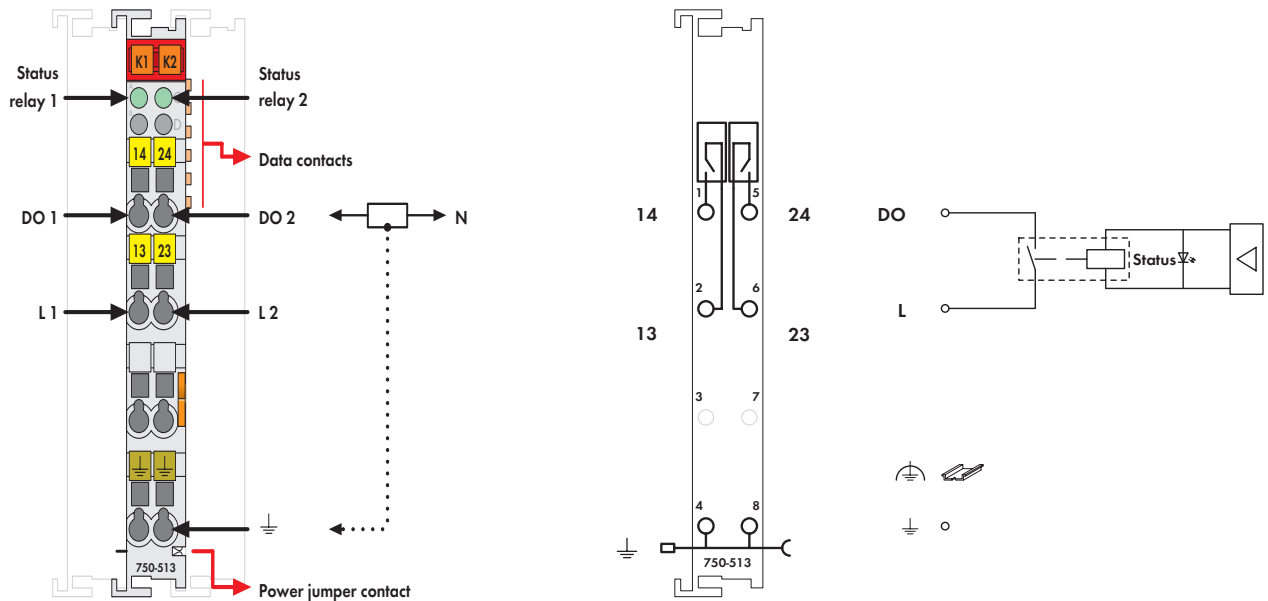


Fig. 750 Series/Technical data see page 24/Delivered without miniature WSB markers
750/753 Series marking see pages 10 ... 11 / 12 ... 13

The connected load is switched via the digital output (relay contacts) from the control system.

The internal system voltage is used to trigger the relay.

The NO contacts are electrically isolated.

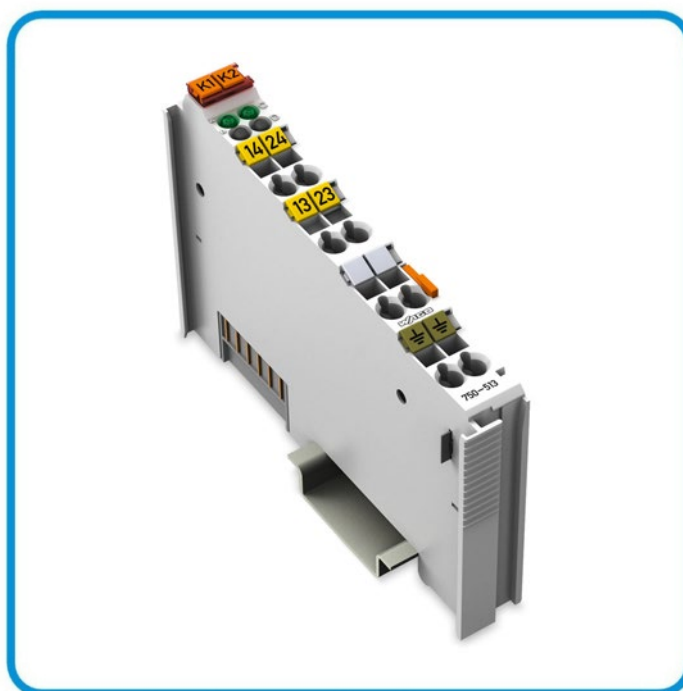
The switched status of the relay is shown by a LED.

Actuators with a ground (earth) wire may be directly connected to the module.

Description	Item No.	Pack. Unit
2DO 230V AC 2.0A/ Relay 2NO/ Potential Free	750-513	10 ¹⁾
2DO 230V AC 2.0A/ Relay 2NO/ Potential Free / without power jumper	750-513/000-001	1
2DO 230V AC 2.0A/ Relay 2NO/ Potential Free (without connector)	753-513	1
¹⁾ Also available individually		
Accessories	Item No.	Pack. Unit
753 Series Connectors	753-110	25
Coding elements	753-150	100
Miniature WSB Quick marking system		
plain	248-501	5
with marking	see pages 352 ... 353	
Approvals	Also see "Approvals Overview" in Section 1	
Conformity marking	CE	
Shipbuilding (versions upon request)	ABS, BV, DNV, GL, KR, LR*, NKK*, PRS*, RINA* *753 Series, pending	
UL 508		
ANSI/ISA 12.12.01	Class I, Div. 2, Grp. ABCD, T4	
EN 60079-0, -15	I M2 / II 3 GD Ex nC IIC T4	
EN 61241-0, -1		

Technical Data	
No. of outputs	2 make contacts
Max. current consumption (internal)	100 mA
Max. switching voltage	250 V AC / 30 V DC
Switching power	500 VA / 60 W (resistive load) cos φ max. = 0.4; L/R max = 7 ms
Min. switching current	10 mA / 5 V DC
Max. switching current	2 A AC/DC
Max. switching frequency	30/min (at nominal load)
Pull-in time (max.)	10 ms
Bounce time (typ.)	1.2 ms
Drop-out time (max.)	10 ms
Contact material	Silver alloy
Mechanical life	2 × 10 ⁷ switching operations
Electrical life	3 × 10 ⁵ switching operations (2 A / 250 V AC) or (2 A / 30 V DC)
Isolation	1.5 kV eff. (field/system)*; * 2.5 kV rated surge voltage; Overvoltage category III
Internal bit width	2 bits
Wire connection	CAGE CLAMP®
Cross sections	0.08 mm ² ... 2.5 mm ² / AWG 28 ... 14
Stripped lengths, 750/753 Series	8 ... 9 mm / 0.33 in 9 ... 10 mm / 0.37 in
Width	12 mm
Weight	53.5 g
EMC: CE - immunity to interference	acc. to EN 60000-6-2 (2005)
EMC: CE - emission of interference	acc. to EN 61000-6-4 (2007)
EMC: marine applications	
- immunity to interference	acc. to Germanischer Lloyd (2003)
EMC: marine applications	
- emission of interference	acc. to Germanischer Lloyd (2003)

This Page Left Intentionally Blank



WAGO-I/O-SYSTEM 750
2DO 230V AC 2.0A/ Relay 2NO/Potential Free
750-513(/xxx-xxx)
2-Channel Relay Output Module 230 V AC, 30 V DC,
Isolated outputs; 2 make contacts

Version 1.2.0



© 2014 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

Table of Contents	3
1 Notes about this Documentation	5
1.1 Validity of this Documentation	5
1.2 Copyright.....	5
1.3 Symbols.....	6
1.4 Number Notation.....	8
1.5 Font Conventions	8
2 Important Notes	9
2.1 Legal Bases	9
2.1.1 Subject to Changes	9
2.1.2 Personnel Qualifications.....	9
2.1.3 Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions	9
2.1.4 Technical Condition of Specified Devices.....	10
2.2 Safety Advice (Precautions).....	11
3 Device Description	13
3.1 View	15
3.2 Connectors.....	16
3.2.1 Data Contacts/Internal Bus.....	16
3.2.2 Power Jumper Contacts/Field Supply	17
3.2.3 CAGE CLAMP® Connectors	18
3.3 Display Elements	18
3.4 Schematic Diagram	19
3.5 Technical Data	20
3.5.1 Device Data	20
3.5.2 Power Supply.....	20
3.5.3 Communication	20
3.5.4 Outputs	20
3.5.5 Connection Type	21
3.5.6 Climatic Environmental Conditions.....	21
3.5.7 Load Limiting Value Graph of the Relay.....	22
3.6 Approvals	23
3.7 Standards and Guidelines	25
4 Process Image	26
5 Mounting	27
5.1 Mounting Sequence.....	27
5.2 Inserting and Removing Devices	28
5.2.1 Inserting the I/O Module	28
5.2.2 Removing the I/O Module.....	29
6 Connect Devices	30
6.1 Connecting a Conductor to the CAGE CLAMP®	30
6.2 Connection Example	31
6.3 Protective Circuits for Contacts of Relay Modules.....	32

7	Use in Hazardous Environments	34
7.1	Marking Configuration Examples	35
7.1.1	Marking for Europe According to ATEX and IEC-Ex	35
7.1.2	Marking for America According to NEC 500	40
7.2	Installation Regulations	41
7.2.1	Special Conditions for Safe Use (ATEX Certificate TÜV 07 ATEX 554086 X)	42
7.2.2	Special Conditions for Safe Use (ATEX Certificate TÜV 12 ATEX 106032 X)	43
7.2.3	Special Conditions for Safe Use (IEC-Ex Certificate TUN 09.0001 X)	44
7.2.4	Special Conditions for Safe Use (IEC-Ex Certificate IECEx TUN 12.0039 X)	45
7.2.5	Special Conditions for Safe Use according to ANSI/ISA 12.12.01 ...	46
	List of Figures	47
	List of Tables.....	48

1 Notes about this Documentation

Note



Always retain this documentation!

This documentation is part of the product. Therefore, retain the documentation during the entire service life of the product. Pass on the documentation to any subsequent user. In addition, ensure that any supplement to this documentation is included, if necessary.

1.1 Validity of this Documentation

This documentation is only applicable to the I/O module 750-513 (2DO 230V AC 2.0A/ Relay 2NO/Potential Free) and the variants listed in the table below.

Table 1: Variants

Item Number/Variant	Designation
750-513	2DO 230V AC 2.0A/ Relay 2NO/Potential Free
750-513/000-001	2DO 230V AC 2.0A/ Relay 2NO/Potential Free/ without power jumper contacts

Note



Documentation Validity for Variants

Unless otherwise indicated, the information given in this documentation applies to listed variants.

The I/O module 750-513 shall only be installed and operated according to the instructions in this manual and in the manual for the used fieldbus coupler/controller.

NOTICE

Consider power layout of the WAGO-I/O-SYSTEM 750!

In addition to these operating instructions, you will also need the manual for the used fieldbus coupler/controller, which can be downloaded at www.wago.com. There, you can obtain important information including information on electrical isolation, system power and supply specifications.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.3 Symbols

 **DANGER****Personal Injury!**

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **DANGER****Personal Injury Caused by Electric Current!**

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **WARNING****Personal Injury!**

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

 **CAUTION****Personal Injury!**

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE**Damage to Property!**

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE**Damage to Property Caused by Electrostatic Discharge (ESD)!**

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note**Important Note!**

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.

Information



Additional Information:

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.4 Number Notation

Table 2: Number Notation

Number Code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.5 Font Conventions

Table 3: Font Conventions

Font Type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Programme\WAGO-I/O-CHECK</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualifications

All sequences implemented on WAGO-I/O-SYSTEM 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

All changes to the coupler or controller should always be carried out by qualified personnel with sufficient skills in PLC programming.

2.1.3 Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions

Fieldbus couplers, fieldbus controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to actuators or higher-level control systems. Using programmable controllers, the signals can also be (pre-) processed.

The devices have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the devices in wet and dusty environments is prohibited.

Operating the WAGO-I/O-SYSTEM 750 devices in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section “Device Description” > “Standards and Guidelines” in the manual for the used fieldbus coupler/controller.

Appropriate housing (per 94/9/EG) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

2.1.4 Technical Condition of Specified Devices

The devices to be supplied ex works are equipped with hardware and software configurations, which meet the individual application requirements. WAGO Kontakttechnik GmbH & Co. KG will be exempted from any liability in case of changes in hardware or software as well as to non-compliant usage of devices.

Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

2.2 Safety Advice (Precautions)

For installing and operating purposes of the relevant device to your system the following safety precautions shall be observed:



DANGER

Do not work on devices while energized!

All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.

DANGER

Install the device only in appropriate housings, cabinets or in electrical operation rooms!

The WAGO-I/O-SYSTEM 750 and its components are an open system. As such, install the system and its components exclusively in appropriate housings, cabinets or in electrical operation rooms. Allow access to such equipment and fixtures to authorized, qualified staff only by means of specific keys or tools.

NOTICE

Replace defective or damaged devices!

Replace defective or damaged device/module (e.g., in the event of deformed contacts), since the long-term functionality of device/module involved can no longer be ensured.

NOTICE

Protect the components against materials having seeping and insulating properties!

The components are not resistant to materials having seeping and insulating properties such as: aerosols, silicones and triglycerides (found in some hand creams). If you cannot exclude that such materials will appear in the component environment, then install the components in an enclosure being resistant to the above-mentioned materials. Clean tools and materials are imperative for handling devices/modules.

NOTICE

Clean only with permitted materials!

Clean soiled contacts using oil-free compressed air or with ethyl alcohol and leather cloths.

NOTICE**Do not use any contact spray!**

Do not use any contact spray. The spray may impair contact area functionality in connection with contamination.

NOTICE**Do not reverse the polarity of connection lines!**

Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

NOTICE**Avoid electrostatic discharge!**

The devices are equipped with electronic components that may be destroyed by electrostatic discharge when touched. Please observe the safety precautions against electrostatic discharge per DIN EN 61340-5-1/-3. When handling the devices, please ensure that environmental factors (personnel, work space and packaging) are properly grounded.

3 Device Description

The 750-513 Digital Output Module transmits binary control signals from the automation device to the connected actuators (e.g., solenoid valves, contactors, transmitters, relays or other electrical loads).

The I/O module has 2 output channels, providing a direct connection to actuators with ground contact.

The actuators can be wired directly to the DO 1 terminal and to an external N potential, or to DO 2 and to an external N potential. An external potential of 230 V AC or 30 V DC can be applied to relay contacts L1 and L2.

The assignment of the connections is described in the “Connectors” section. Connection examples are shown in section “Connecting Devices” > ... > “Connection Example(s)”.

NOTICE

Limit all induced voltage!

The electronic components of the I/O module can be damaged by the induced voltage produced when inductive loads are de-activated.

An appropriate protection circuit, e.g., a recovery diode, must be installed in parallel to the load to limit this induced voltage.

The I/O module contains two make-contact, potential-free relays. This means that only the connecting points for the relay contacts are available at the field side so that different voltages can be applied independently to both make contacts.

The floating contacts also enable, for example, series connection of both relays, or series connection of several relays when the I/O modules are set up in blocks.

When a relay is closed, the potential applied at L1 or L2 is routed to the associated second relay contact DO1 or DO2.

The internal system voltage is used to trigger the relays.

The switching status of the two relays is indicated via green status LEDs. The meaning of the LEDs is described in the “Display Elements” section.

Note



Use the supply module!

The standard version of the 750-513 I/O module is equipped with an integrated power contact for relaying the ground potential, while the version 750-513/000-001 does not come equipped with a power contact. When required, use a power supply module for the field-side supply of the following I/O modules.

With consideration of the power jumper contacts, the individual modules can be arranged in any combination when configuring the fieldbus node. An arrangement in groups within the group of potentials is not necessary.

NOTICE

Do not exceed maximum current via power jumper contacts!

The maximum current to flow through the power jumper contacts is 10 A. Greater currents can damage the contacts.

When configuring your system, ensure that this current is not exceeded. If exceeded, insert an additional supply module.

The I/O module 750-513 can be used with all fieldbus couplers/controllers of the WAGO-I/O-SYSTEM 750.

3.1 View

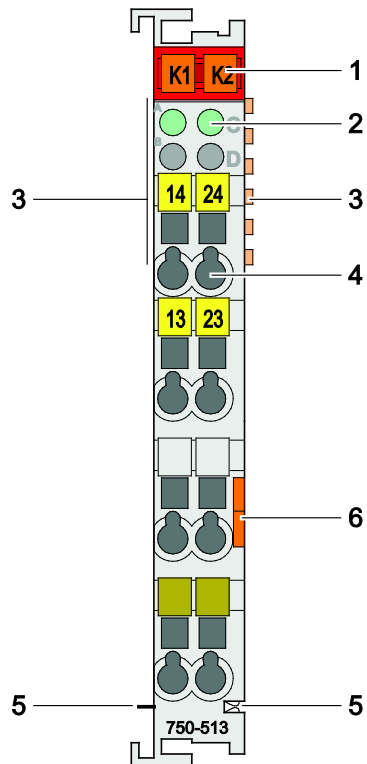


Figure 1: View

Table 4: Legend for Figure “View”

Pos.	Description	Details See Section
1	Marking possibility with Mini-WSB	---
2	Status LEDs	“Device Description” > “Display Elements”
3	Data contacts	“Device Description” > “Connectors”
4	CAGE CLAMP® connectors	“Device Description” > “Connectors”
5	Power jumper contacts	“Device Description” > “Connectors”
6	Release tab	“Mounting” > “Inserting and Removing Devices”

3.2 Connectors

3.2.1 Data Contacts/Internal Bus

Communication between the fieldbus coupler/controller and the I/O modules as well as the system supply of the I/O modules is carried out via the internal bus. It is comprised of 6 data contacts, which are available as self-cleaning gold spring contacts.

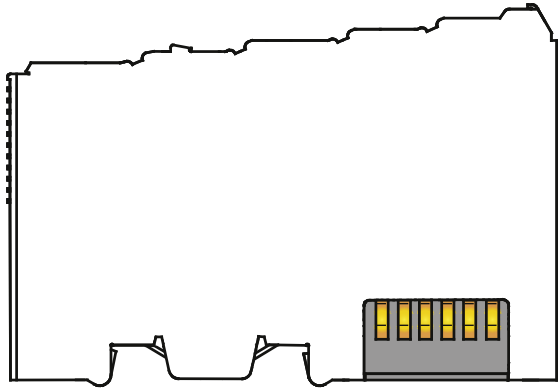


Figure 2: Data Contacts

NOTICE

Do not place the I/O modules on the gold spring contacts!

Do not place the I/O modules on the gold spring contacts in order to avoid soiling or scratching!

NOTICE



Ensure that the environment is well grounded!

The devices are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the devices, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, e.g. data contacts.

3.2.2 Power Jumper Contacts/Field Supply

Note



Use the supply module!

The standard version of the 750-513 I/O module is equipped with an integrated power contact for relaying the ground potential, while the version 750-513/000-001 does not come equipped with a power contact. When required, use a power supply module for the field-side supply of the following I/O modules.

CAUTION

Risk of injury due to sharp-edged blade contacts!

The blade contacts are sharp-edged. Handle the I/O module carefully to prevent injury.

The I/O module 750-513 has 1 self-cleaning power jumper contact that supply and transmit power for the field side. The contact on the left side of the I/O module is designed as male contact and the contact on the right side as spring contact.

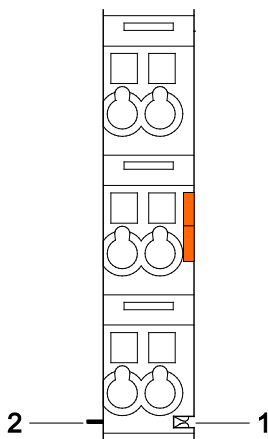


Figure 3: Power Jumper Contacts

Table 5: Legend for Figure “Power Jumper Contacts”

Contact	Type	Function
1	Spring contact	Potential transmission (Earth) for field supply
2	Blade contact	Potential feed-in (Earth) for field supply

NOTICE

Do not exceed maximum current via power jumper contacts!

The maximum current to flow through the power jumper contacts is 10 A. Greater currents can damage the contacts.

When configuring your system, ensure that this current is not exceeded. If exceeded, insert an additional supply module.

3.2.3 CAGE CLAMP® Connectors

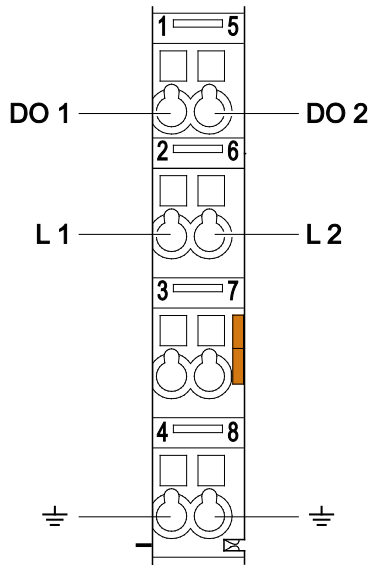


Figure 4: CAGE CLAMP® Connectors

Table 6: Legend for Figure “CAGE CLAMP® Connectors“

Channel	Designation	Connector	Function
1	DO 1	1	Relay contact DO 1
	L1	2	Relay contact L1
	Earth	4	Field supply earth
2	DO 2	5	Relay contact DO 2
	L2	6	Relay contact L2
	Earth	8	Field supply earth

3.3 Display Elements

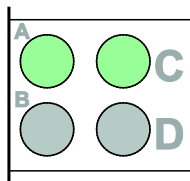


Figure 5: Display Elements

Table 7: Legend for Figure “Display Elements”

Channel	Designation	LED	State	Function
1	State relay 1	A	Off	Make contact DO1 opened
			Green	Make contact DO1 closed
2	State relay 2	C	Off	Make contact DO2 opened
			Green	Make contact DO2 closed

3.4 Schematic Diagram

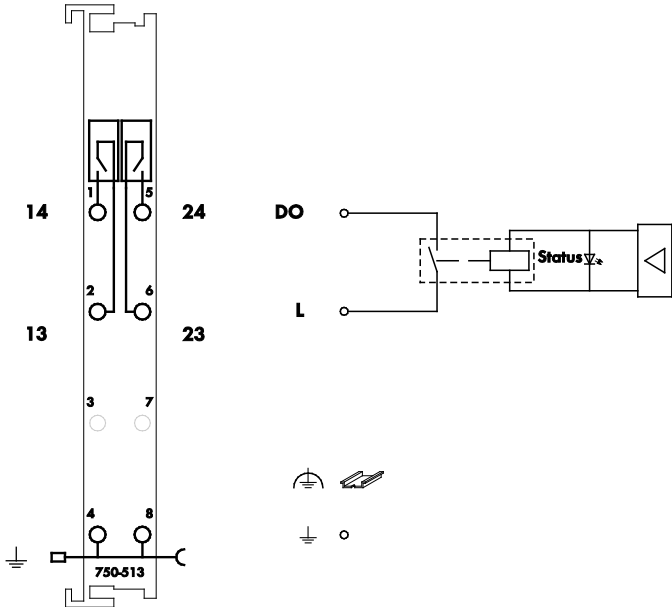


Figure 6: Schematic Diagram

3.5 Technical Data

3.5.1 Device Data

Table 8: Technical Data – Device

Width	12 mm
Height (from upper edge of 35 DIN rail)	64 mm
Depth	100 mm
Weight	Approx. 55 g

3.5.2 Power Supply

Table 9: Technical Data – Power Supply

Current consumption (internal)	100 mA
Switching voltage	250 VAC / 30 VDC
Switching power	500 VA/60 W $\cos\phi_{\max} = 0.4$; $L/R_{\max} = 7$ ms
Switching current _{min.}	10 mA / 5 VDC
Switching current _{max.}	2 A AC/DC
Isolation	1.5 kV eff. (field/system)
Rated surge voltage	2.5 kV
Overvoltage category	III

3.5.3 Communication

Table 10: Technical Data – Communication

Internal bit width	2 bits out
--------------------	------------

3.5.4 Outputs

Table 11: Technical Data – Outputs

No. of outputs	2 make contacts
Switching frequency _{max.}	30/min (at nominal load)
Pull-in time _{max.}	10 ms
Bounce time _{typ.}	1.2 ms
Drop-out time _{max.}	10 ms
Contact material	Silver alloy
Mechanical life _{max.}	2×10^7 switching operations
Electrical life _{min.}	3×10^5 (2 A / 250 VAC) or (2 A / 30 V DC)

3.5.5 Connection Type

Table 12: Technical Data – Field Wiring

Wire connection	CAGE CLAMP®
Cross section	0.08 mm ² ... 2.5 mm ² , AWG 28 ... 14
Stripped lengths	8 mm ... 9 mm / 0.33 in

Table 13: Technical Data – Power Jumper Contacts

Power jumper contacts	Blade/spring contact, self-cleaning
Voltage drop at I _{max.}	< 1 V/64 modules

Table 14: Technical Data – Data Contacts

Data contacts	Slide contact, hard gold plated, self-cleaning
---------------	--

3.5.6 Climatic Environmental Conditions

Table 15: Technical Data – Climatic Environmental Conditions

Operating temperature range	0 °C ... 55 °C
Storage temperature range	-25 °C ... +85 °C
Relative humidity without condensation	Max. 95 %
Resistance to harmful substances	Acc. to IEC 60068-2-42 and IEC 60068-2-43
Maximum pollutant concentration at relative humidity < 75 %	SO ₂ ≤ 25 ppm H ₂ S ≤ 10 ppm
Special conditions	Ensure that additional measures for components are taken, which are used in an environment involving: – dust, caustic vapors or gases – ionizing radiation

3.5.7 Load Limiting Value Graph of the Relay

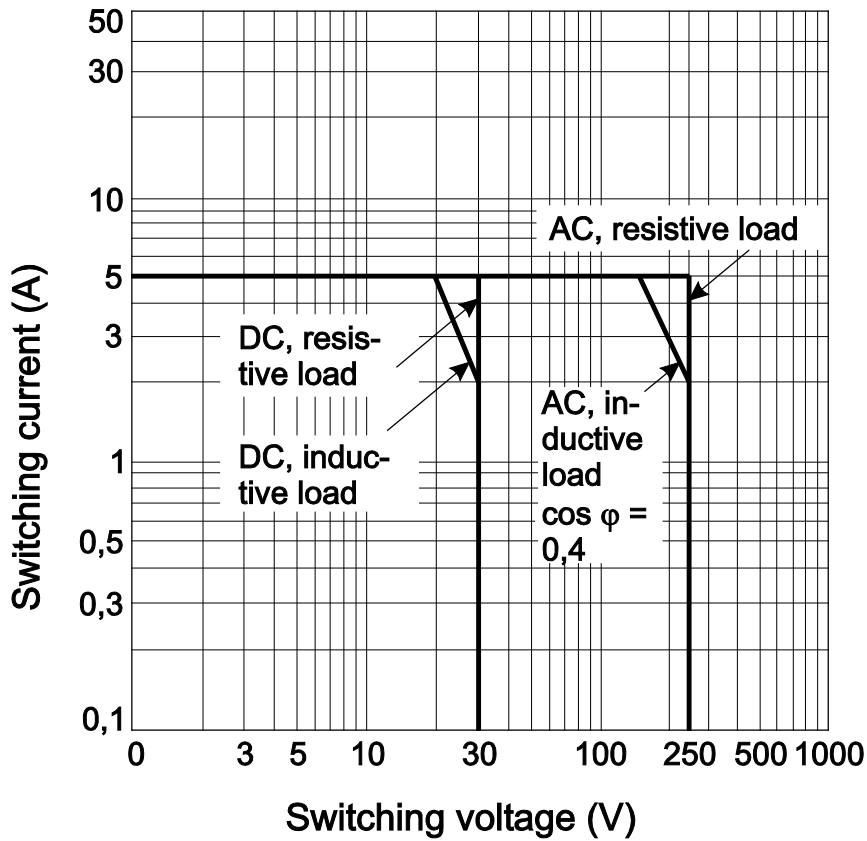


Figure 7: Load Limiting Value Graph of the Relay

3.6 Approvals

Information



More information about approvals.

Detailed references to the approvals are listed in the document “Overview Approvals **WAGO-I/O-SYSTEM 750**”, which you can find via the internet under: www.wago.com > SERVICES > DOWNLOADS > Additional documentation and information on automation products > WAGO-I/O-SYSTEM 750 > System Description.

The following approvals have been granted to the basic version and all variants of 750-513 I/O modules:



Conformity Marking



cUL_{US}

UL508



Korea Certification

MSIP-REM-W43-DOM750

The following approvals have been granted to the basic version of 750-513 I/O modules and the variation 750-513/000-001:



TÜV 07 ATEX 554086 X

I M2 Ex d I Mb
II 3 G Ex nA nC IIC T4 Gc
II 3 D Ex tc IIIC T135°C Dc

Ambient temperature range:

$0\text{ °C} \leq T_a \leq +60\text{ °C}$

IECEX TUN 09.0001 X

Ex d I Mb
Ex nA nC IIC T4 Gc
Ex tc IIIC T135°C Dc

Ambient temperature range:

$0\text{ °C} \leq T_a \leq +60\text{ °C}$













cUL_{US}

ANSI/ISA 12.12.01

Class I, Div2 ABCD T4

The following ship approvals have been granted to the basic version and all variants of 750-513 I/O modules listed in the table:

Table 16: Ship Approvals

										
750-513	X	X	X	X	X	X	X	X	X	X
750-513 /000-001		X	X	X	X	X	X		X	X



ABS (American Bureau of Shipping)



Federal Maritime and Hydrographic Agency



BV (Bureau Veritas)



DNV (Det Norske Veritas) Class B



GL (Germanischer Lloyd) Cat. A, B, C, D (EMC 1)



KR (Korean Register of Shipping)



LR (Lloyd's Register) Env. 1, 2, 3, 4



NKK (Nippon Kaiji Kyokai)



PRS (Polski Rejestr Statków)



RINA (Registro Italiano Navale)

3.7 Standards and Guidelines

All variations of 750-513 I/O modules meet the following requirements on emission and immunity of interference:

EMC CE-Emission of interference	acc. to EN 61000-6-4: 2007
EMC CE-Immunity to interference	acc. to EN 61000-6-2: 2005
EMC marine applications-Emission of interference	acc. to Germanischer Lloyd (2003)
EMC marine applications-Immunity to interference	acc. to Germanischer Lloyd (2003)

4 Process Image

Note



Mapping of process data in the process image of fieldbus systems

The representation of the process data of some I/O modules or their variants in the process image depends on the fieldbus coupler/controller used. Please take this information from the section “Fieldbus Specific Design of the Process Data” included in the description concerning the process image of the corresponding fieldbus coupler/controller.

Table 17: Output Bits

		Bit 1	Bit 0
		DO 2	DO 1
DO 1	Signal state DO 1 – digital output channel 1		
DO 2	Signal state DO 2 – digital output channel 2		

5 Mounting

5.1 Mounting Sequence

Fieldbus couplers/controllers and I/O modules of the WAGO-I/O-SYSTEM 750 are snapped directly on a carrier rail in accordance with the European standard EN 50022 (DIN 35).

The reliable positioning and connection is made using a tongue and groove system. Due to the automatic locking, the individual devices are securely seated on the rail after installation.

Starting with the fieldbus coupler/controller, the I/O modules are mounted adjacent to each other according to the project design. Errors in the design of the node in terms of the potential groups (connection via the power contacts) are recognized, as the I/O modules with power contacts (blade contacts) cannot be linked to I/O modules with fewer power contacts.

CAUTION

Risk of injury due to sharp-edged blade contacts!

The blade contacts are sharp-edged. Handle the I/O module carefully to prevent injury.

NOTICE

Insert I/O modules only from the proper direction!

All I/O modules feature grooves for power jumper contacts on the right side. For some I/O modules, the grooves are closed on the top. Therefore, I/O modules featuring a power jumper contact on the left side cannot be snapped from the top. This mechanical coding helps to avoid configuration errors, which may destroy the I/O modules. Therefore, insert I/O modules only from the right and from the top.

Note



Don't forget the bus end module!

Always plug a bus end module 750-600 onto the end of the fieldbus node! You must always use a bus end module at all fieldbus nodes with WAGO-I/O-SYSTEM 750 fieldbus couplers/controllers to guarantee proper data transfer.

5.2 Inserting and Removing Devices

NOTICE

Perform work on devices only if they are de-energized!

Working on energized devices can damage them. Therefore, turn off the power supply before working on the devices.

5.2.1 Inserting the I/O Module

1. Position the I/O module so that the tongue and groove joints to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are engaged.

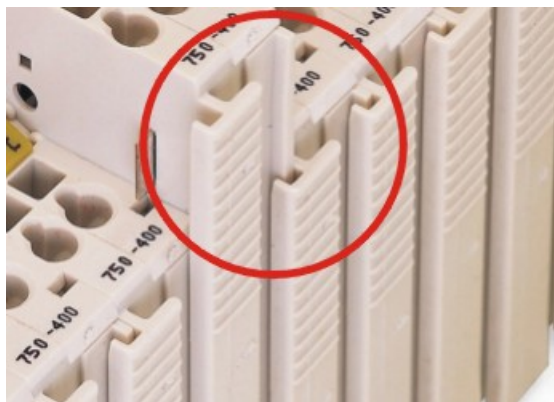


Figure 8: Insert I/O Module (Example)

2. Press the I/O module into the assembly until the I/O module snaps into the carrier rail.

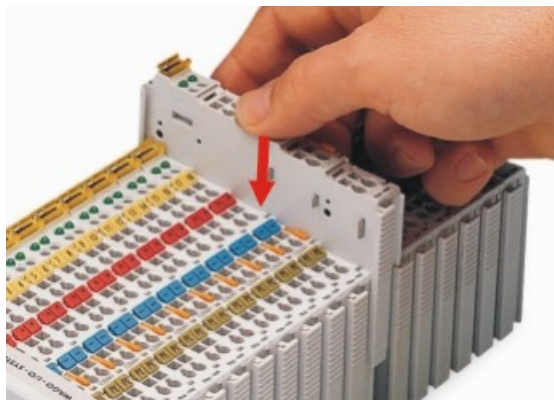


Figure 9: Snap the I/O Module into Place (Example)

With the I/O module snapped in place, the electrical connections for the data contacts and power jumper contacts (if any) to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are established.

5.2.2 Removing the I/O Module

1. Remove the I/O module from the assembly by pulling the release tab.

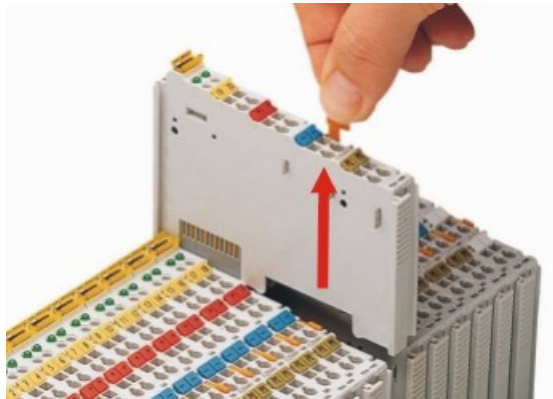


Figure 10: Removing the I/O Module (Example)

Electrical connections for data or power jumper contacts are disconnected when removing the I/O module.

6 Connect Devices

6.1 Connecting a Conductor to the CAGE CLAMP®

The WAGO CAGE CLAMP® connection is appropriate for solid, stranded and finely stranded conductors.

Note



Only connect one conductor to each CAGE CLAMP®!

Only one conductor may be connected to each CAGE CLAMP®.

Do not connect more than one conductor at one single connection!

If more than one conductor must be routed to one connection, these must be connected in an up-circuit wiring assembly, for example using WAGO feed-through terminals.

Exception:

If it is unavoidable to jointly connect 2 conductors, then you must use a ferrule to join the wires together. The following ferrules can be used:

Length:	8 mm
Nominal cross section _{max.} :	1 mm ² for 2 conductors with 0.5 mm ² each
WAGO product:	216-103 or products with comparable properties

1. For opening the CAGE CLAMP® insert the actuating tool into the opening above the connection.
2. Insert the conductor into the corresponding connection opening.
3. For closing the CAGE CLAMP® simply remove the tool. The conductor is now clamped firmly in place.

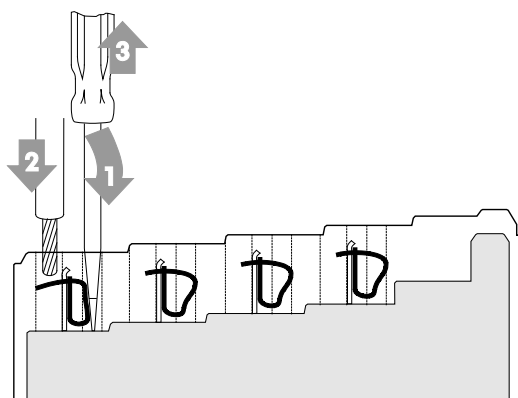


Figure 11: Connecting a Conductor to a CAGE CLAMP®

6.2 Connection Example

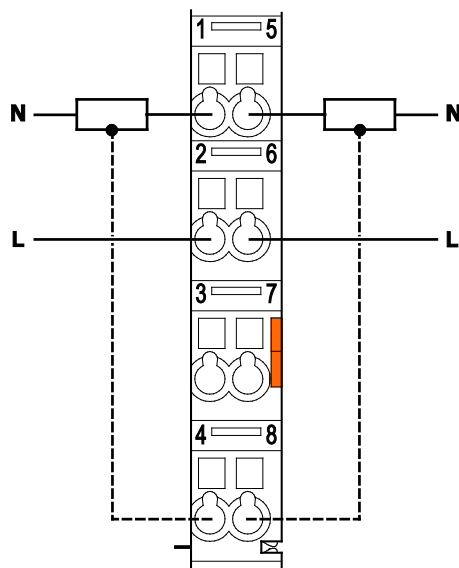


Figure 12: Connecting Diagram

6.3 Protective Circuits for Contacts of Relay Modules

Switching off inductive loads such as contactors and solenoid valves can generate transients with voltage peaks of up to several thousand volts. Very often, these transients exceed the permissible limits specified in the EMC standards. They must therefore be limited by using external protective circuits.

Furthermore, they cause an arc, which may destroy the switching contact, and has a significant impact on the lifetime and safety of the relay. The following protective circuits, which are directly connected to the source of the interference in parallel with the consumer, have proved successful in practice:

Table 18: Protective Circuits for Contacts of Relay Modules

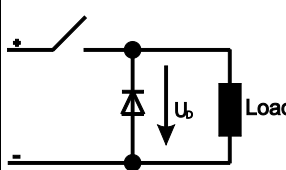
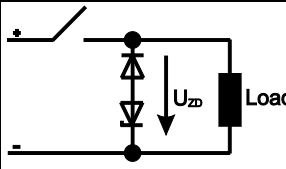
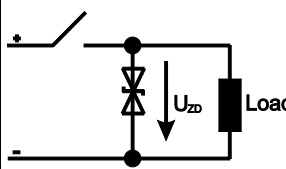
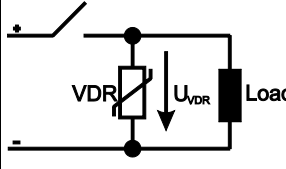
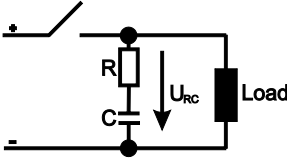
Protective Circuits	Additional Off-Delay	Defined Induction Voltage Limitation	Bipolar Effective Attenuation	Advantages and Disadvantages
 <p>Figure 13: Diode</p>	long	yes (U_D)	no	Advantages: <ul style="list-style-type: none"> easy implementation cost-effective and reliable uncritical dimensioning low induction voltages Disadvantages: <ul style="list-style-type: none"> attenuation only via load resistor long off-delay
 <p>Figure 14: Zener Diode and Diode</p>	medium to short	yes (U_{ZD})	no	Advantages: <ul style="list-style-type: none"> uncritical dimensioning Disadvantages: <ul style="list-style-type: none"> attenuation only above U_{ZD}
 <p>Figure 15: Suppressor Diode</p>	medium to short	yes (U_{ZD})	yes	Advantages: <ul style="list-style-type: none"> cost-effective uncritical dimensioning limitation of positive and negative peaks suitable for AC voltage Disadvantages: <ul style="list-style-type: none"> attenuation only above U_{ZD}
 <p>Figure 16: Varistor</p>	medium to short	yes (U_{VDR})	yes	Advantages: <ul style="list-style-type: none"> high power absorption uncritical dimensioning suitable for AC voltage Disadvantages: <ul style="list-style-type: none"> attenuation only above U_{VDR}

Table 18: Protective Circuits for Contacts of Relay Modules

Protective Circuits	Additional Off-Delay	Defined Induction Voltage Limitation	Bipolar Effective Attenuation	Advantages and Disadvantages
 <p>Figure 17: R/C Combination</p>	medium to short	no	yes	<p>Advantages:</p> <ul style="list-style-type: none"> • HF attenuation via power storage • suitable for AC voltage • level-independent attenuation <p>Disadvantages:</p> <ul style="list-style-type: none"> • exact dimensioning required • high inrush current

7 Use in Hazardous Environments

The **WAGO-I/O-SYSTEM 750** (electrical equipment) is designed for use in Zone 2 hazardous areas.

The following sections include both the general identification of components (devices) and the installation regulations to be observed. The individual subsections of the “Installation Regulations” section must be taken into account if the I/O module has the required approval or is subject to the range of application of the ATEX directive.

7.1 Marking Configuration Examples

7.1.1 Marking for Europe According to ATEX and IEC-Ex

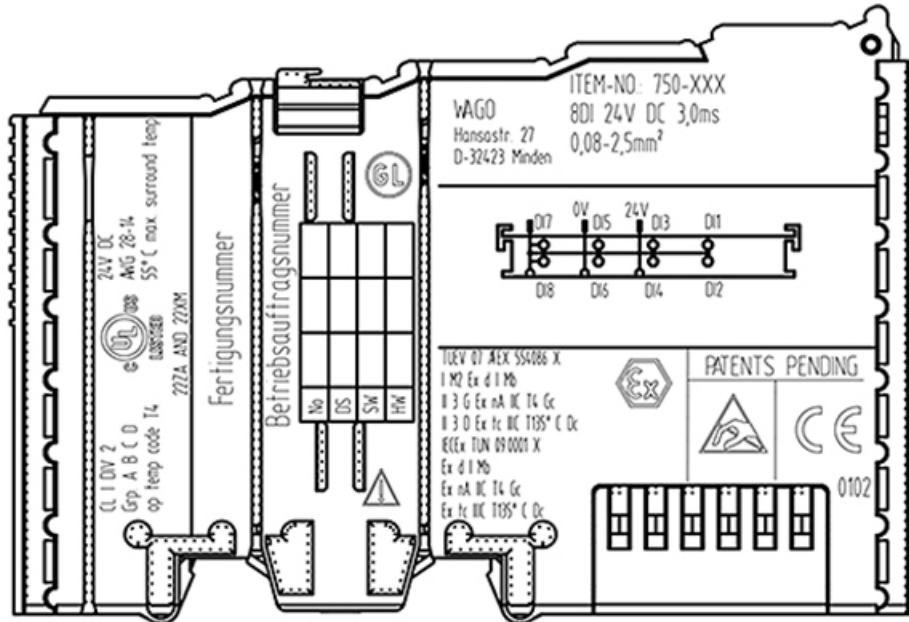


Figure 18: Side Marking Example for Approved I/O Modules According to ATEX and IECEx

TUEV 07 AEX 554086 X
I M2 Ex d I Mb
II 3 G Ex nA IIC T4 Gc
II 3 D Ex tc IIC T135° C Dc
IECEx TUN 09.0001 X
Ex d I Mb
Ex nA IIC T4 Gc
Ex tc IIC T135° C Dc



Figure 19: Text Detail – Marking Example for Approved I/O Modules According to ATEX and IECEx.

Table 19: Description of Marking Example for Approved I/O Modules According to ATEX and IECEx

Printing on Text	Description
TÜV 07 ATEX 554086 X IECEx TUN 09.0001 X	Approving authority and certificate numbers
Dust	
II	Equipment group: All except mining
3D	Category 3 (Zone 22)
Ex	Explosion protection mark
tc Dc	Type of protection and equipment protection level (EPL): protection by enclosure
IIIC	Explosion group of dust
T 135°C	Max. surface temperature of the enclosure (without a dust layer)
Mining	
I	Equipment group: Mining
M2	Category: High level of protection
Ex	Explosion protection mark
d Mb	Type of protection and equipment protection level (EPL): Flameproof enclosure
I	Explosion group for electrical equipment for mines susceptible to firedamp
Gases	
II	Equipment group: All except mining
3G	Category 3 (Zone 2)
Ex	Explosion protection mark
nA Gc	Type of protection and equipment protection level (EPL): Non-sparking equipment
nC Gc	Type of protection and equipment protection level (EPL): Sparking apparatus with protected contacts. A device which is so constructed that the external atmosphere cannot gain access to the interior
IIIC	Explosion group of gas and vapours
T4	Temperature class: Max. surface temperature 135°C

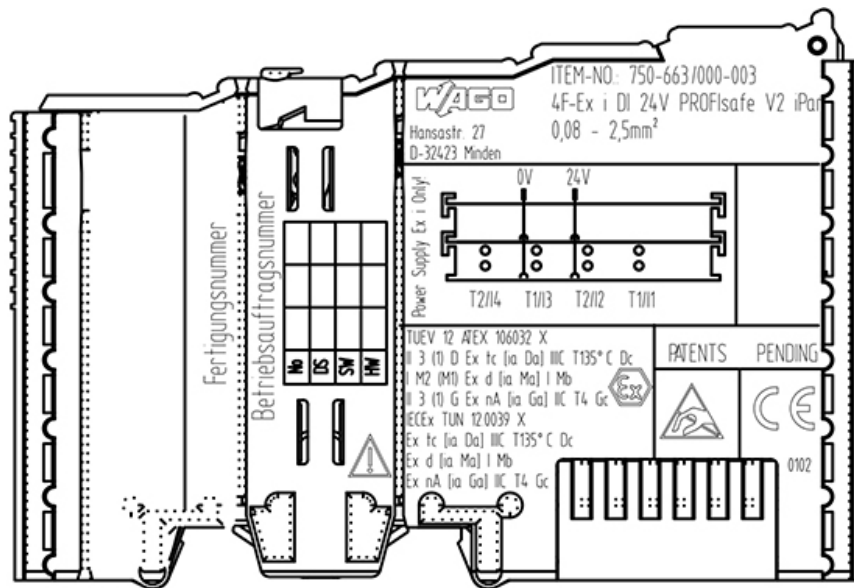


Figure 20: Side Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx.


TUEV 12 ATEX 106032 X
 II 3 (1) D Ex tc [ia Da] IIC T135° C Dc
 I M2 (M1) Ex d [ia Ma] I Mb
 II 3 (1) G Ex nA [ia Ga] IIC T4 Gc 
 IECEX TUN 12.0039 X
 Ex tc [ia Da] IIC T135° C Dc
 Ex d [ia Ma] I Mb
 Ex nA [ia Ga] IIC T4 Gc

Figure 21: Text Detail – Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx.

Table 20: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx

Inscription Text	Description
TÜV 07 ATEX 554086 X IECEX TUN 09.0001X	Approving authority and certificate numbers
TÜV 12 ATEX 106032 X IECEX TUN 12.0039 X	
Dust	
II	Equipment group: All except mining
3(1)D	Category 3 (Zone 22) equipment containing a safety device for a category 1 (Zone 20) equipment
3(2)D	Category 3 (Zone 22) equipment containing a safety device for a category 2 (Zone 21) equipment
Ex	Explosion protection mark
tc Dc	Type of protection and equipment protection level (EPL): protection by enclosure
[ia Da]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 20
[ib Db]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 21
IIIC	Explosion group of dust
T 135°C	Max. surface temperature of the enclosure (without a dust layer)
Mining	
I	Equipment Group: Mining
M2 (M1)	Category: High level of protection with electrical circuits which present a very high level of protection
Ex d Mb	Explosion protection mark with Type of protection and equipment protection level (EPL): Flameproof enclosure
[ia Ma]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety electrical circuits
I	Explosion group for electrical equipment for mines susceptible to firedamp

Table 20: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx

Gases	
II	Equipment group: All except mining
3(1)G	Category 3 (Zone 2) equipment containing a safety device for a category 1 (Zone 0) equipment
3(2)G	Category 3 (Zone 2) equipment containing a safety device for a category 2 (Zone 1) equipment
Ex	Explosion protection mark
nA Gc	Type of protection and equipment protection level (EPL): Non-sparking equipment
[ia Ga]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 0
[ia Gb]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 1
IIC	Explosion group of gas and vapours
T4	Temperature class: Max. surface temperature 135°C

7.1.2 Marking for America According to NEC 500

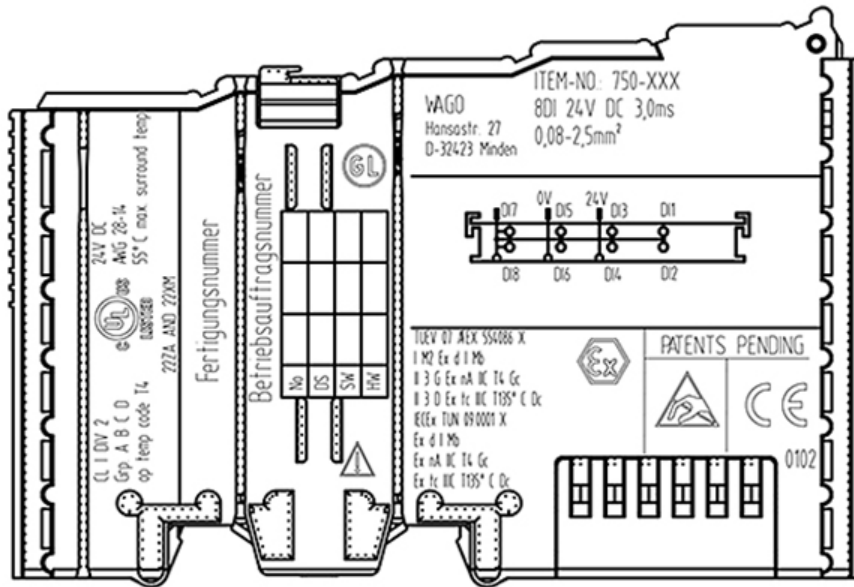


Figure 22: Side Marking Example for I/O Modules According to NEC 500

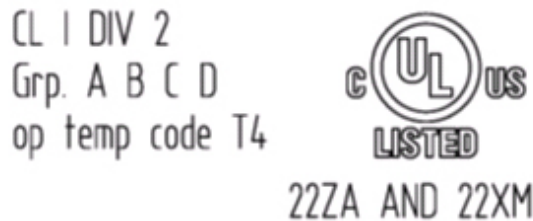


Figure 23: Text Detail – Marking Example for Approved I/O Modules According to NEC 500

Table 21: Description of Marking Example for Approved I/O Modules According to NEC 500

Printing on Text	Description
CL I	Explosion protection group (condition of use category)
DIV 2	Area of application
Grp. ABCD	Explosion group (gas group)
Op temp code T4	Temperature class

7.2 Installation Regulations

For the installation and operation of electrical equipment in hazardous areas, the valid national and international rules and regulations which are applicable at the installation location must be carefully followed.

7.2.1 Special Conditions for Safe Use (ATEX Certificate TÜV 07 ATEX 554086 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15 and EN 60079-31. For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. Dip-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “Memory-Card”, “USB”, “Fieldbus connection”, “Configuration and programming interface”, “antenna socket”, “D-Sub”, “DVI-port” and the “Ethernet interface”. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.
6. For replaceable fuses the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The following warnings shall be placed nearby the unit:
WARNING – DO NOT REMOVE OR REPLACE FUSE WHEN ENERGIZED
WARNING – DO NOT SEPARATE WHEN ENERGIZED
WARNING – SEPARATE ONLY IN A NON-HAZARDOUS AREA

7.2.2 Special Conditions for Safe Use (ATEX Certificate TÜV 12 ATEX 106032 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** Ex i shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15 and EN 60079-31. For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
4. For the type the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.

7.2.3 Special Conditions for Safe Use (IEC-Ex Certificate TUN 09.0001 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15 and IEC 60079-31. For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64.
The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “Memory-Card”, “USB”, “Fieldbus connection”, “Configuration and programming interface”, “antenna socket”, “D-Sub”, “DVI-port” and the “Ethernet interface”. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.
6. For replaceable fuses the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The following warnings shall be placed nearby the unit:
WARNING – DO NOT REMOVE OR REPLACE FUSE WHEN ENERGIZED
WARNING – DO NOT SEPARATE WHEN ENERGIZED
WARNING – SEPARATE ONLY IN A NON-HAZARDOUS AREA

7.2.4 Special Conditions for Safe Use (IEC-Ex Certificate IECEx TUN 12.0039 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus independent I/O Modules WAGO-I/O-SYSTEM 750-*** Ex i shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15, IEC 60079-31.
For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64.
The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
4. For the type the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.

7.2.5 Special Conditions for Safe Use according to ANSI/ISA 12.12.01

- A. “This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only.”
- B. “This equipment is to be fitted within tool-secured enclosures only.”
- C. “WARNING Explosion hazard - substitution of components may impair suitability for Class I, Div. 2.”
- D. “WARNING – Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous” has to be placed near each operator accessible connector and fuse holder.
- E. When a fuse is provided, the following information shall be provided: “A switch suitable for the location where the equipment is installed shall be provided to remove the power from the fuse.”
- F. For devices with EtherCAT/Ethernet connectors “Only for use in LAN, not for connection to telecommunication circuits.”
- G. “WARNING - Use Module 750-642 only with antenna module 758-910.”
- H. For Couplers/Controllers and Economy bus modules only: The instructions shall contain the following: “The configuration interface Service connector is for temporary connection only. Do not connect or disconnect unless the area is known to be non-hazardous. Connection or disconnection in an explosive atmosphere could result in an explosion.”
- I. Modules containing fuses only: “WARNING - Devices containing fuses must not be fitted into circuits subject to over loads, e.g. motor circuits.”
- J. Modules containing SD card reader sockets only: “WARNING - Do not connect or disconnect SD-Card while circuit is live unless the area is known to be free of ignitable concentrations of flammable gases or vapors.”

Information



Additional Information

Proof of certification is available on request.

Also take note of the information given on the operating and assembly instructions.

The manual, containing these special conditions for safe use, must be readily available to the user.

List of Figures

Figure 1: View	15
Figure 2: Data Contacts	16
Figure 3: Power Jumper Contacts	17
Figure 4: CAGE CLAMP [®] Connectors	18
Figure 5: Display Elements	18
Figure 6: Schematic Diagram	19
Figure 7: Load Limiting Value Graph of the Relay	22
Figure 8: Insert I/O Module (Example)	28
Figure 9: Snap the I/O Module into Place (Example)	28
Figure 10: Removing the I/O Module (Example)	29
Figure 11: Connecting a Conductor to a CAGE CLAMP [®]	30
Figure 12: Connecting Diagram	31
Figure 13: Diode	32
Figure 14: Zener Diode and Diode	32
Figure 15: Suppressor Diode	32
Figure 16: Varistor	32
Figure 17: R/C Combination	33
Figure 18: Side Marking Example for Approved I/O Modules According to ATEX and IECEx	35
Figure 19: Text Detail – Marking Example for Approved I/O Modules According to ATEX and IECEx	35
Figure 20: Side Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx	37
Figure 21: Text Detail – Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx	37
Figure 22: Side Marking Example for I/O Modules According to NEC 500	40
Figure 23: Text Detail – Marking Example for Approved I/O Modules According to NEC 500	40

List of Tables

Table 1: Variants	5
Table 2: Number Notation.....	8
Table 3: Font Conventions	8
Table 4: Legend for Figure “View”	15
Table 5: Legend for Figure “Power Jumper Contacts”	17
Table 6: Legend for Figure “CAGE CLAMP® Connectors“	18
Table 7: Legend for Figure “Display Elements”	18
Table 8: Technical Data – Device	20
Table 9: Technical Data – Power Supply.....	20
Table 10: Technical Data – Communication.....	20
Table 11: Technical Data – Outputs.....	20
Table 12: Technical Data – Field Wiring.....	21
Table 13: Technical Data – Power Jumper Contacts	21
Table 14: Technical Data – Data Contacts	21
Table 15: Technical Data – Climatic Environmental Conditions	21
Table 16: Ship Approvals.....	24
Table 17: Output Bits	26
Table 18: Protective Circuits for Contacts of Relay Modules.....	32
Table 19: Description of Marking Example for Approved I/O Modules According to ATEX and IECEx	36
Table 20: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx	38
Table 21: Description of Marking Example for Approved I/O Modules According to NEC 500.....	40

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: +49/5 71/8 87 – 0
Fax: +49/5 71/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>

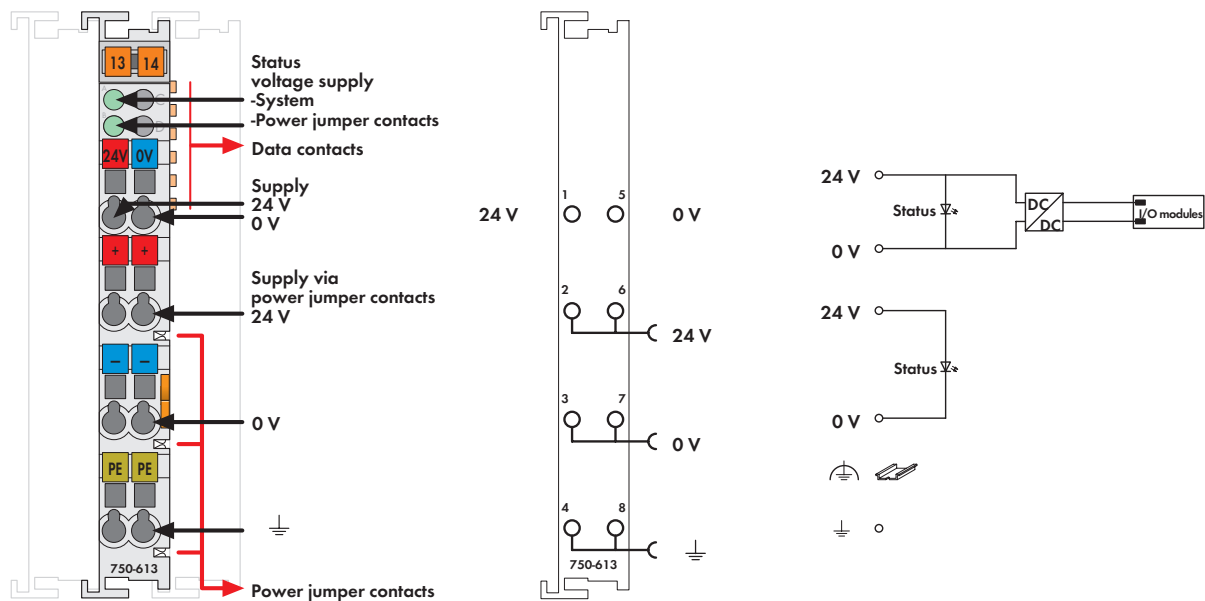


This Page Left Intentionally Blank

WAGO SYSTEM 750-613
INTERNAL SYSTEM POWER
SUPPLY MODULE

Operations & Maintenance Manual
December 2015

Internal System Supply Module 24 V DC



The internal system supply module increases the current supply for the internal system by 2A.

If the internal current consumption of all modules is higher than 2A, an additional supply module must be added.

The module also supplies field side power to the adjoining modules via the power jumper contacts.

Description	Item No.	Pack. Unit
24V DC Bus Power Supply	750-613	10 ¹⁾
¹⁾ Also available individually		
Accessories	Item No.	Pack. Unit
Miniature WSB Quick marking system		
plain	248-501	5
with marking	see pages 352 ... 353	
Approvals	Also see "Approvals Overview" in Section 1	
Conformity marking	CE	
Shipbuilding	ABS, BV, DNV, GL, KR, LR, NKK, PRS, RINA	
UL 508		
ANSI/ISA 12.12.01	Class I, Div. 2, Grp. ABCD, T4	
EN 60079-0, -15	I M2 / II 3 GD Ex nA IIC T4	
EN 61241-0, -1		

Technical Data	
Power supply	24 V DC (-15 % ... +20 %)
Input current max.	500 mA
Total current for I/O modules	2000 mA
Voltage via power jumper contacts (max.)	24 V DC (-25 % ... +30 %)
Current via power jumper contacts (max.)	10 A DC
Wire connection	CAGE CLAMP®
Cross sections	0.08 mm ² ... 2.5 mm ² / AWG 28 ... 14
Stripped lengths	8 ... 9 mm / 0.33 in
Width	12 mm
Weight	58.5 g
EMC: CE - immunity to interference	acc. to EN 61000-6-2 (2005)
EMC: CE - emission of interference	acc. to EN 61000-6-4 (2007)
EMC: marine applications	
- immunity to interference	acc. to Germanischer Lloyd (2003)
EMC: marine applications	
- emission of interference	acc. to Germanischer Lloyd (2003)

This Page Left Intentionally Blank

WAGO → I/O → SYSTEM 750

**Fieldbus Independent
I/O Modules**

**24 V DC Bus Power Supply
750-613**



Manual

Version 1.0.3

Copyright © 2006 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0

Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55

Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the correctness and completeness of this documentation. However, as errors can never be fully excluded, we would appreciate any information or ideas at any time.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally trademark or patent protected.

CONTENT

1 Important Comments	4
1.1 Legal Principles	4
1.1.1 Copyright	4
1.1.2 Personnel Qualification	4
1.1.3 Intended Use	4
1.2 Symbols	5
1.3 Number Notation	5
1.4 Safety Notes	6
1.5 Scope	6
2 I/O Modules	7
2.1 System Modules	7
2.1.1 750-613 [24 V DC Bus Power Supply]	7
2.1.1.1 View	7
2.1.1.2 Description	7
2.1.1.3 Display Elements	8
2.1.1.4 Schematic Diagram	9
2.1.1.5 Technical Data	10
2.1.1.6 Power supply concept	12

1 Important Comments

To ensure fast installation and start-up of the units described in this manual, we strongly recommend that the following information and explanations are carefully read and abided by.

1.1 Legal Principles

1.1.1 Copyright

This manual is copyrighted, together with all figures and illustrations contained therein. Any use of this manual which infringes the copyright provisions stipulated herein, is not permitted. Reproduction, translation and electronic and photo-technical archiving and amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG. Non-observance will entail the right of claims for damages.

WAGO Kontakttechnik GmbH & Co. KG reserves the right to perform modifications allowed by technical progress. In case of grant of a patent or legal protection of utility patents all rights are reserved by WAGO Kontakttechnik GmbH & Co. KG. Products of other manufacturers are always named without referring to patent rights. The existence of such rights can therefore not be ruled out.

1.1.2 Personnel Qualification

The use of the product detailed in this manual is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the valid standards. WAGO Kontakttechnik GmbH & Co. KG declines all liability resulting from improper action and damage to WAGO products and third party products due to non-observance of the information contained in this manual.

1.1.3 Intended Use

For each individual application, the components supplied are to work with a dedicated hardware and software configuration. Modifications are only permitted within the framework of the possibilities documented in the manuals. All other changes to the hardware and/or software and the non-conforming use of the components entail the exclusion of liability on part of WAGO Kontakttechnik GmbH & Co. KG.

Please direct any requirements pertaining to a modified and/or new hardware or software configuration directly to WAGO Kontakttechnik GmbH & Co. KG.

1.2 Symbols



Danger

Always abide by this information to protect persons from injury.



Warning

Always abide by this information to prevent damage to the device.



Attention

Marginal conditions must always be observed to ensure smooth operation.



ESD (Electrostatic Discharge)

Warning of damage to the components by electrostatic discharge. Observe the precautionary measure for handling components at risk.



Note

Routines or advice for efficient use of the device and software optimization.



More information

References on additional literature, manuals, data sheets and INTERNET pages

1.3 Number Notation

Number Code	Example	Note
Decimal	100	normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	Within ', Nibble separated with dots

1.4 Safety Notes



Warning

Switch off the system prior to working on bus modules!

In the event of deformed contacts, the module in question is to be replaced, as its functionality can no longer be ensured on a long-term basis.

The components are not resistant against materials having seeping and insulating properties. Belonging to this group of materials is: e.g. aerosols, silicones, triglycerides (found in some hand creams).

If it cannot be ruled out that these materials appear in the component environment, then additional measures are to be taken:

- installation of the components into an appropriate enclosure
 - handling of the components only with clean tools and materials.
-



Attention

Cleaning of soiled contacts may only be done with ethyl alcohol and leather cloths. Thereby, the ESD information is to be regarded.

Do not use any contact spray. The spray may impair the functioning of the contact area.

The WAGO-I/O-SYSTEM 750 and its components are an open system. It must only be assembled in housings, cabinets or in electrical operation rooms. Access must only be given via a key or tool to authorized qualified personnel.

The relevant valid and applicable standards and guidelines concerning the installation of switch boxes are to be observed.



ESD (Electrostatic Discharge)

The modules are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the modules, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, e.g. gold contacts.

1.5 Scope

This manual describes the System Module **750-613 24 V DC Bus Power Supply** of the modular WAGO-I/O-SYSTEM 750.

Handling, assembly and start-up are described in the manual of the Fieldbus Coupler. Therefore this documentation is valid only in the connection with the appropriate manual.

2 I/O Modules

2.1 System Modules

2.1.1 750-613 [24 V DC Bus Power Supply]

Internal System Supply Module DC 24 V

2.1.1.1 View

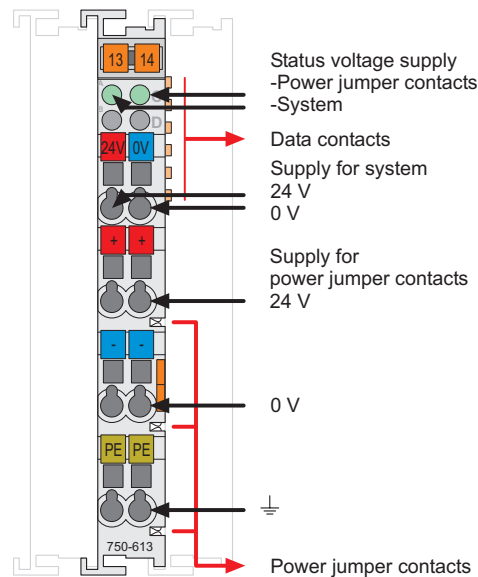


Fig. 2.1.1-1: Supply Module 750-613

g061300e

2.1.1.2 Description

The supply module 750-613 provides the internal system power to the downstream modules and increases the system current supply by 2 A. The system power supply is connected to the upper 24 V / 0 V terminals in parallel to the coupler / controller.



Attention

Resetting the system by switching on and off the system supply must take place simultaneously for all supply modules (coupler / controller and internal system supply module 750-613)!

Furthermore the supply module 750-613 provides DC 24 V field side power to the downstream modules.

The field power supply is derived from an external source via the lower 24 V, 0V and ground (earth) terminals.

Two green status LEDs indicate the 24 V voltage supply (system and field power supply).

Both the potential groups and the individual modules within these groups can be arranged in any combination when designing the field bus node.

The supply voltage for the field side is made automatically through the individual I/O modules by means of power jumper contacts.



Warning

The maximum current of the supply module is 10 A. When configuring the system it is important not to exceed the maximum/sum current. However, if such a case should occur, another supply module must be added.

The supply module 750-613 can be used with all couplers/controllers of the WAGO-I/O-SYSTEM 750.



Attention

The supplementary power supply regulations (chapter 0, “

Power supply concept”) must be observed for the certified operation of the supply module 750-613 in shipbuilding or offshore and onshore applications!

2.1.1.3 Display Elements

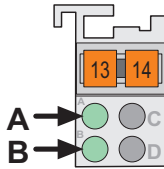


Fig. 2.1.1-2: Display Elements g062602x

or

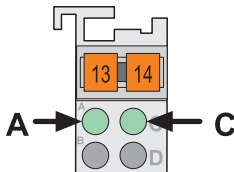


Fig. 2.1.1-3: Display Elements g062602x

LED	Designation	State	Function
A green	Status voltage supply – System	off	No DC 24 V system voltage supply.
		on	DC 24 V system voltage supply.
B or C ^{*)} green	Status voltage supply – Power jumper contacts.	off	No DC 24 V voltage supply via power jumper contacts.
		on	DC 24 V voltage supply via power jumper contacts.

^{*)} LED position is manufacturing dependent

2.1.1.4 Schematic Diagram

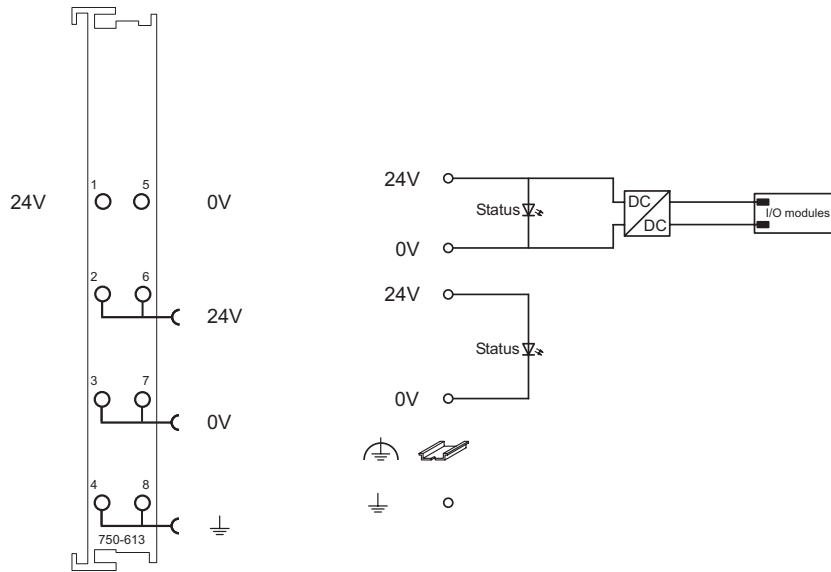













Fig. 2.1.1-4: Supply Module 750-613

g061301e

2.1.1.5 Technical Data

Module Specific Data		
Voltage Supply	DC 24 V (-25% ... +30%)	
Input current _{max.}	500 mA	
Total current for I/O Modules	2000 mA	
Voltage via power jumper contacts _{max.}	DC 24 V (-25% ... +30%)	
Current via power jumper contacts _{max.}	10 A	
Dimensions W x H* x L * from upper edge of 35 DIN rail	12 mm x 64 mm x 100 mm	
Weight _{ca.}	60 g	
Standards and Regulations (cf. Chapter 2.2 of the Coupler/Controller Manual)		
EMC-Immunity to interference (CE)	acc. to EN 50082-2 (1996)	
EMC-Emission of interference (CE)	acc. to EN 50081-1 (1993)	
EMC-Immunity to interference (Ship building)	acc. to Germanischer Lloyd (97)	
EMC-Emission of interference (Ship building)	acc. to Germanischer Lloyd (97)	
Approvals (cf. Chapter 2.2 of the Coupler/Controller Manual)		
	cUL _{US} (UL508)	
	ABS (American Bureau of Shipping) ¹⁾	
	BV (Bureau Veritas) ¹⁾	
	DNV (Det Norske Veritas) ¹⁾	Cl. B
	GL (Germanischer Lloyd) ¹⁾	Cat. A, B, C, D
	KR (Korean Register of Shipping) ¹⁾	
	LR (Lloyd's Register) ¹⁾	Env. 1, 2, 3, 4
	NKK (Nippon Kaiji Kyokai)	
	RINA (Registro Italiano Navale) ¹⁾	
	cUL _{US} (UL1604)	Class I Div2 ABCD T4A
	KEMA	II 3 G EEx nA II T4
	Conformity Marking	

¹⁾ Note information on "Voltage Supply"!



More Information

Detailed references to the approvals are listed in the document "Overview Approvals WAGO-I/O-SYSTEM 750", which you can find on the CD ROM ELECTRONICC Tools and Docs (Item-No.: 0888-0412)

or in the internet under:

www.wago.com → Documentation → WAGO-I/O-SYSTEM 750 → System Description

2.1.1.6 Power supply concept

The system power supply is fed in via the upper 24 V / 0 V terminals in parallel to the coupler / controller and is protected using a joint fuse.

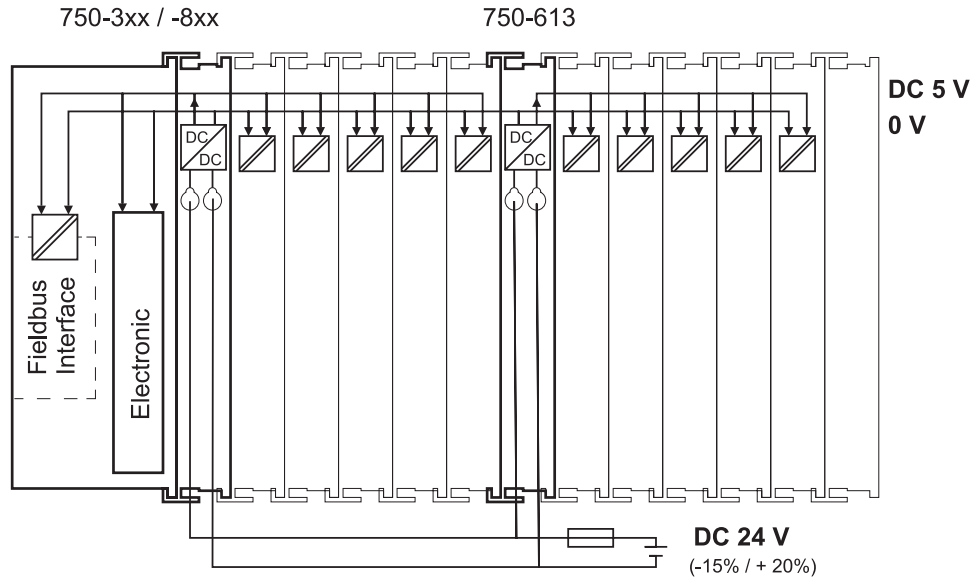


Fig. 2.1.1-5: System supply with a coupler / controller

g0xxx06e

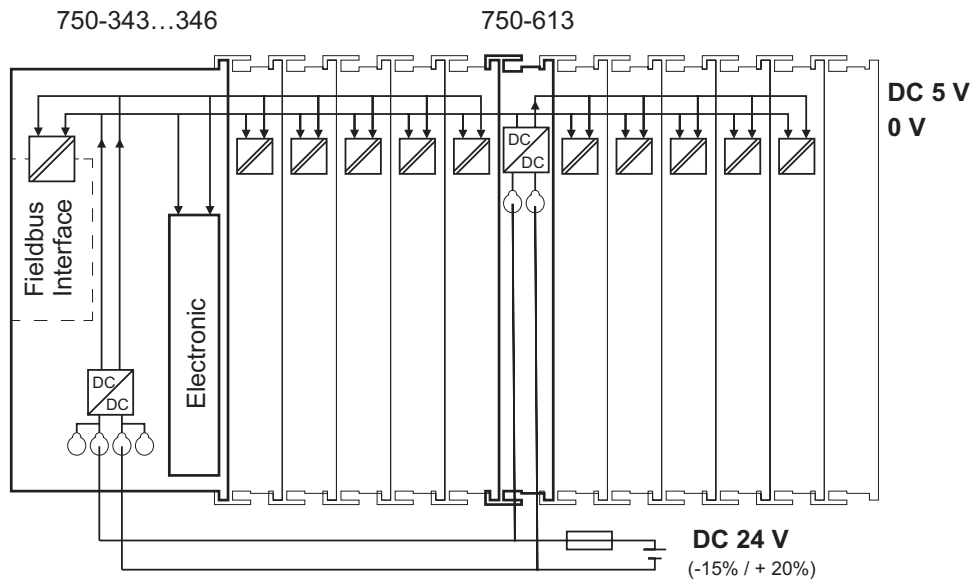


Fig. 2.1.1-6: System supply with an ECO-coupler

g0xxx12e



Attention

Resetting the system by switching on and off the system supply must take place simultaneously for all supply modules (coupler / controller and internal system supply module 750-613)

Supplementary power supply regulations:

The overvoltage protection module 750-626, which is equipped with surge suppression for the 24 V DC field side power supply and used to filter the 24 V DC field power supply, is required for the certified operation of the supply module 750-613 in shipbuilding or offshore and onshore applications.

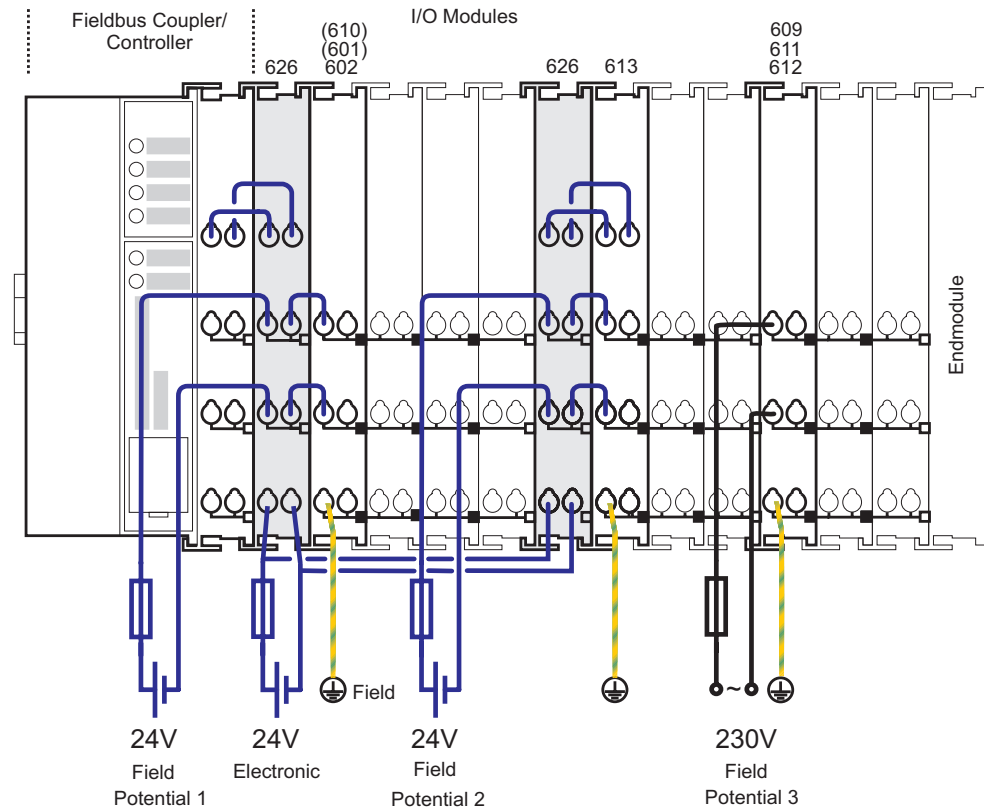


Abb. 2.1.1-7: Power supply concept

g061303e



WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: 05 71/8 87 – 0
Fax: 05 71/8 87 – 1 69
E-Mail: info@wago.com

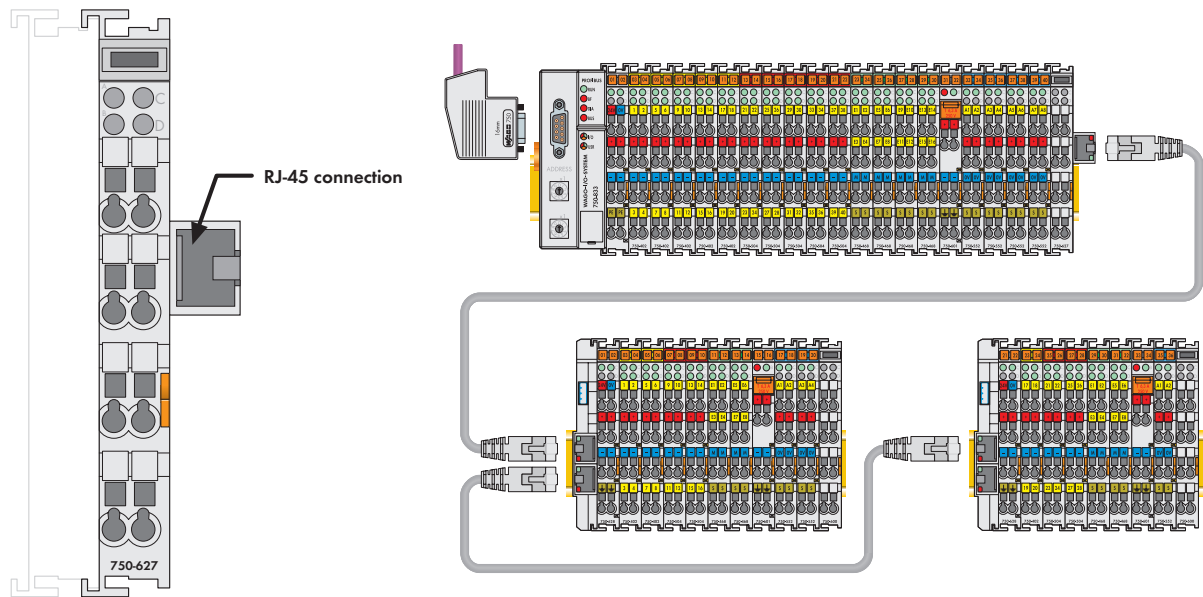
Internet: <http://www.wago.com>

This Page Left Intentionally Blank

WAGO SYSTEM 750-627
INTERNAL DATA BUS
EXTENSION END MODULE

Operations & Maintenance Manual
December 2015

Internal Data Bus Extension End Module



Delivered without miniature WSB markers


The end module for the internal data bus extension 750-627 is attached to the end of the I/O terminal block like the standard end module 750-600. The block is terminated with the module, to which a connecting cable can be attached with an RJ-45 connector.

Power to the internal electronics is supplied via the internal bus. Together with at least one coupler module for the internal data bus extension 750-628 the module forms a functional unit. The fieldbus coupler/controller carries out all diagnosis and commissioning tasks.

Installation note Attention:

To ensure safe, reliable operating states when using the internal data bus extension 750-627/-628 these states must be registered prior to startup with the following couplers or PLCs (refer to manual for supported couplers/PLCs). You must use the "WAGO Extension Setting" software for this (download: www.wago.com).

Please complete the manufacturing number matrix on the right-hand side of the couplers when updating the firmware and internal operating parameters.

Description	Item No.	Pack. Unit
Internal Data Bus Extension End Module	750-627	1
Accessories		
Software „WAGO Extension Setting“	Download: www.wago.com	
Communication cable (used to register or remove the end extension module)	750-920	10
Miniature WSB Quick marking system		
	plain	248-501 5
	with marking	see pages 352 ... 353
Approvals		
	Also see "Approvals Overview" in Section 1	
Conformity marking	CE	
Shipbuilding	GL	
UL 508		

Technical Data	
Max. no. of coupler modules	up to 10
Max. current consumption (internal)	70 mA
Buscoupler connection	1 x RJ-45 socket
Distance	max. 5 m (end module and coupler module)
Transmission medium	shielded copper wire (ETHERNET patch cable) 4 x 2 x 0.25 mm ² , twisted pair, double shielding
Isolation	500 V system/supply
Wire connection	CAGE CLAMP®
Cross sections	0.08 mm ² ... 2.5 mm ² / AWG 28 ... 14
Stripped lengths	8 ... 9 mm / 0.33 in
Width	24 mm
Weight	45.5 g
EMC: CE - immunity to interference	acc. to EN 61000-6-2 (2005)
EMC: CE - emission of interference	acc. to EN 61000-6-4 (2007)

This Page Left Intentionally Blank



WAGO-I/O-SYSTEM 750
Module Bus Extension End Module
750-627

Version 1.1.0



© 2014 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

1	Notes about this Documentation.....	5
1.1	Validity of this Documentation.....	5
1.2	Copyright.....	5
1.3	Symbols.....	6
1.4	Number Notation.....	8
1.5	Font Conventions.....	8
2	Important Notes	9
2.1	Legal Bases	9
2.1.1	Subject to Changes	9
2.1.2	Personnel Qualifications.....	9
2.1.3	Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions	9
2.1.4	Technical Condition of Specified Devices.....	10
2.2	Safety Advice (Precautions).....	11
3	Internal Data Bus Structure with Bus Extension	13
4	Device Description	15
4.1	Power Supply	15
4.1.1	Isolation	15
4.2	View	17
4.3	Connectors.....	18
4.3.1	Data Contacts/Internal Bus.....	18
4.3.2	Power Jumper Contacts/Field Supply	19
4.3.3	CAGE CLAMP® Connectors	19
4.3.4	RJ-45 Connectors	19
4.4	Display Elements	19
4.5	Operating Elements	19
4.6	Technical Data	20
4.6.1	Device data	20
4.6.2	Supply.....	20
4.6.3	Connection type.....	20
4.6.4	Communication	20
4.6.5	Climatic Environmental Conditions.....	21
4.7	Approvals	22
4.8	Standards and Guidelines.....	22
5	Mounting.....	23
5.1	Mounting Sequence.....	23
5.2	Inserting and Removing Devices	24
5.2.1	Inserting the I/O Module	24
5.2.2	Removing the I/O Module.....	25
6	Connect Devices	26
6.1	Connecting Cable to RJ-45 Socket	26
7	Use in Hazardous Environments	27
7.1	Marking Configuration Examples.....	28
7.1.1	Marking for Europe According to ATEX and IEC-Ex	28

7.1.2	Marking for America According to NEC 500.....	31
7.2	Installation Regulations.....	32
7.2.1	Special Conditions for Safe Operation of the ATEX and IEC Ex (acc. DEMKO 08 ATEX 142851X and IECEx PTB 07.0064).....	33
7.2.2	Special conditions for safe use (ATEX Certificate TÜV 07 ATEX 554086 X).....	34
7.2.3	Special conditions for safe use (IEC-Ex Certificate TUN 09.0001 X).....	35
7.2.4	Special conditions for safe use (ATEX Certificate DEKRA 11ATEX0203 X).....	36
7.2.5	ANSI/ISA 12.12.01.....	37
	List of Figures.....	39
	List of Tables.....	40

1 Notes about this Documentation



Note

Always retain this documentation!

This documentation is part of the product. Therefore, retain the documentation during the entire service life of the product. Pass on the documentation to any subsequent user. In addition, ensure that any supplement to this documentation is included, if necessary.

1.1 Validity of this Documentation

This documentation is only applicable to the I/O module 750-627 (Module Bus Extension End Module).

The I/O module 750-627 shall only be installed and operated according to the instructions in this manual and in the manual for the used fieldbus coupler/controller.

NOTICE

Consider power layout of the WAGO-I/O-SYSTEM 750!

In addition to these operating instructions, you will also need the manual for the used fieldbus coupler/controller, which can be downloaded at www.wago.com. There, you can obtain important information including information on electrical isolation, system power and supply specifications.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.3 Symbols

DANGER

Personal Injury!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

DANGER



Personal Injury Caused by Electric Current!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

Personal Injury!

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION

Personal Injury!

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE



Damage to Property Caused by Electrostatic Discharge (ESD)!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note



Important Note!

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.



Information

Additional Information:

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.4 Number Notation

Table 1: Number Notation

Number code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.5 Font Conventions

Table 2: Font Conventions

Font type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Programme\WAGO-I/O-CHECK</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualifications

All sequences implemented on WAGO-I/O-SYSTEM 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

All changes to the coupler or controller should always be carried out by qualified personnel with sufficient skills in PLC programming.

2.1.3 Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions

Fieldbus couplers, fieldbus controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to actuators or higher-level control systems. Using programmable controllers, the signals can also be (pre-) processed.

The devices have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the devices in wet and dusty environments is prohibited.

Operating the WAGO-I/O-SYSTEM 750 devices in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section “Device Description” > “Standards and Guidelines” in the manual for the used fieldbus coupler/controller.

Appropriate housing (per 94/9/EG) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

2.1.4 Technical Condition of Specified Devices

The devices to be supplied ex works are equipped with hardware and software configurations, which meet the individual application requirements. WAGO Kontakttechnik GmbH & Co. KG will be exempted from any liability in case of changes in hardware or software as well as to non-compliant usage of devices.

Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

2.2 Safety Advice (Precautions)

For installing and operating purposes of the relevant device to your system the following safety precautions shall be observed:



DANGER

Do not work on devices while energized!

All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.

DANGER

Install the device only in appropriate housings, cabinets or in electrical operation rooms!

The WAGO-I/O-SYSTEM 750 and its components are an open system. As such, install the system and its components exclusively in appropriate housings, cabinets or in electrical operation rooms. Allow access to such equipment and fixtures to authorized, qualified staff only by means of specific keys or tools.

NOTICE

Replace defective or damaged devices!

Replace defective or damaged device/module (e.g., in the event of deformed contacts), since the long-term functionality of device/module involved can no longer be ensured.

NOTICE

Protect the components against materials having seeping and insulating properties!

The components are not resistant to materials having seeping and insulating properties such as: aerosols, silicones and triglycerides (found in some hand creams). If you cannot exclude that such materials will appear in the component environment, then install the components in an enclosure being resistant to the above-mentioned materials. Clean tools and materials are imperative for handling devices/modules.

NOTICE

Clean only with permitted materials!

Clean soiled contacts using oil-free compressed air or with ethyl alcohol and leather cloths.

NOTICE**Do not use any contact spray!**

Do not use any contact spray. The spray may impair contact area functionality in connection with contamination.

NOTICE**Do not reverse the polarity of connection lines!**

Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

**NOTICE****Avoid electrostatic discharge!**

The devices are equipped with electronic components that may be destroyed by electrostatic discharge when touched. Please observe the safety precautions against electrostatic discharge per DIN EN 61340-5-1/-3. When handling the devices, please ensure that environmental factors (personnel, work space and packaging) are properly grounded.

NOTICE**Feed system power supply in parallel!**

The flow of equalizing currents within the bus electronics can destroy the devices. Therefore, connect the system power supply to all supply points of the fieldbus node in parallel. This prevents potential differences and equalizing currents from forming.

NOTICE**Perform insulation tests with direct current (DC)!**

Both the supply voltage and control voltage side are capacitively coupled to the DIN rail. If the modules are mounted on the DIN rail, application of an AC voltage between the two potentials can lead to the destruction of the device.

Use only direct current (DC) for insulation testing.

To avoid destroying the device, discharge the device completely before applying the test voltage again.

3 Internal Data Bus Structure with Bus Extension

With an I/O module 750-627 (end module for bus extension) and one or more I/O modules 750-628 (coupler module for bus extension), a fieldbus node can be divided into max. 11 I/O module blocks (hereinafter referred to as "blocks"). The divided fieldbus nodes still act as a logical unit for the fieldbus coupler/controller. All of the functions of the I/O module system are retained without any changes.

The first block is terminated with an I/O module 750-627 (end module for bus extension). Each subsequent block is started with an I/O module 750-628 (coupler module for bus extension) instead of a fieldbus coupler/controller and terminated with an I/O module 750-600 (end module).

Note



Number of I/O modules independent of bus extension!

The number of total I/O modules permitted in the fieldbus node does not increase by the use of an I/O module for bus extension.

Use an ETHERNET patch cable as a connection cable with the following properties between two blocks:

- Category 5
- Double-insulated shielded twisted pair (STP)
- Cooper
- $4 \times 2 \times 0.25 \text{ mm}^2$

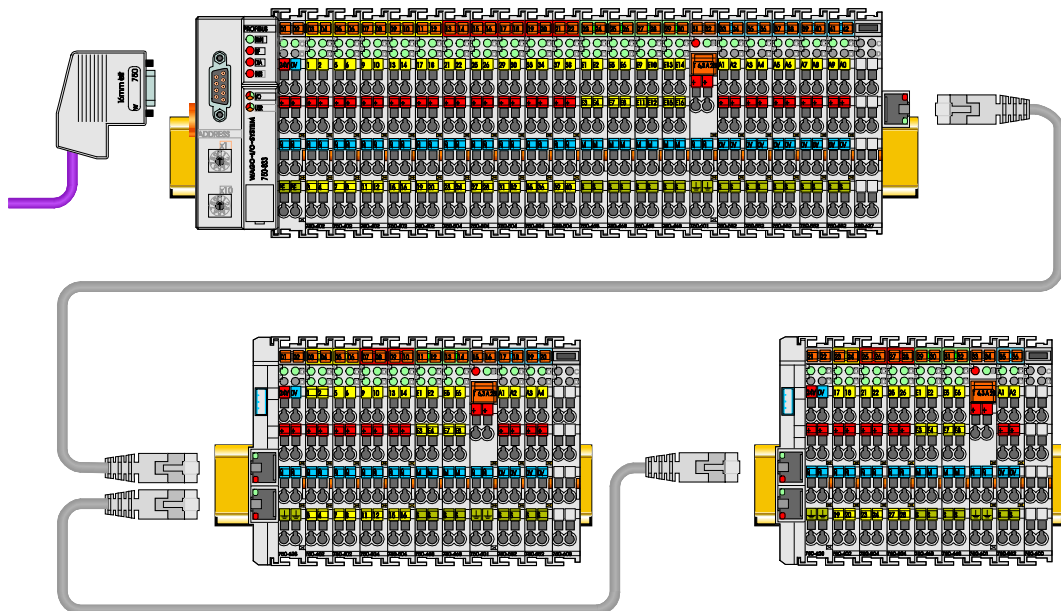


Figure 1: Internal Data Bus Structure with Bus Extension (example)

For the devices as delivered, the cable length between two sub-nodes can be max. 5 meters. Many fieldbus couplers/controllers can be configured, so that the permitted cable length can be increased to max. 10 meters. This setting can be made using the "WAGO Extension Setting Tool" (759-314), which can be downloaded at www.wago.com at no cost.

Note

Note the maximum total cable length of 70 meters!

The length of all connection cables cannot exceed 70 meters.

The maximum cable length permitted cannot be increased for the following fieldbus couplers/controllers:

Item No.	Name
750-300	II/O-LIGHTBUS Fieldbus Coupler
750-301	PROFIBUS Fieldbus Coupler
750-304	INTERBUS Fieldbus Coupler
750-305	CAL Fieldbus Coupler
750-309	LON Fieldbus Coupler
750-310	CC-LINK Fieldbus Coupler
750-312	MODBUS Fieldbus Coupler, RS-485
750-313	SDS Fieldbus Coupler
750-314	MODBUS Fieldbus Coupler, RS-232
750-315	MODBUS Fieldbus Coupler, RS-485
750-316	MODBUS Fieldbus Coupler, RS-232
750-327	CANopen Fieldbus Coupler
750-331	PROFIBUS Fieldbus Coupler
750-339	FireWire Fieldbus Coupler
750-804	INTERBUS Fieldbus Controller
750-812	MODBUS Fieldbus Controller, RS-485
750-814	MODBUS Fieldbus Controller, RS-232
750-815	MODBUS Fieldbus Controller, RS-485
750-816	MODBUS Fieldbus Controller, RS-232

The above table does not include all versions of the fieldbus couplers/controllers. In contrast to standard items, some specific variants may indeed support a configuration change. Corresponding information is given in the documentation for the specific fieldbus coupler/controller.

4 Device Description

The I/O module 750-627 (end module for bus extension) serves in conjunction with the I/O module 750-628 (coupler module for bus extension) to divide a fieldbus node into multiple stations.

The I/O module 750-627 is plugged into the end of the first block instead of the I/O module 750-600 (end module). This station is completed with this module. A cable with RJ-45 connector establishes the connection to the I/O module 750-628 of the next station. More information on the basic structure of a fieldbus node is available in Section "Internal Data Bus Structure with Bus Extension".

The divided fieldbus node still acts as a logical unit for the fieldbus coupler/controller. All of the functions of the I/O module system are retained without any changes.

The internal system supply powers the module.

4.1 Power Supply

4.1.1 Isolation

Within the fieldbus node, there are three electrically isolated potentials:

- Electrically isolated fieldbus interface via transformer
- Electronics of the fieldbus couplers/controllers and the I/O modules (internal bus)
- All I/O modules have an electrical isolation between the electronics (internal bus, logic) and the field electronics. Some digital and analog input modules have each channel electrically isolated, please see catalog.

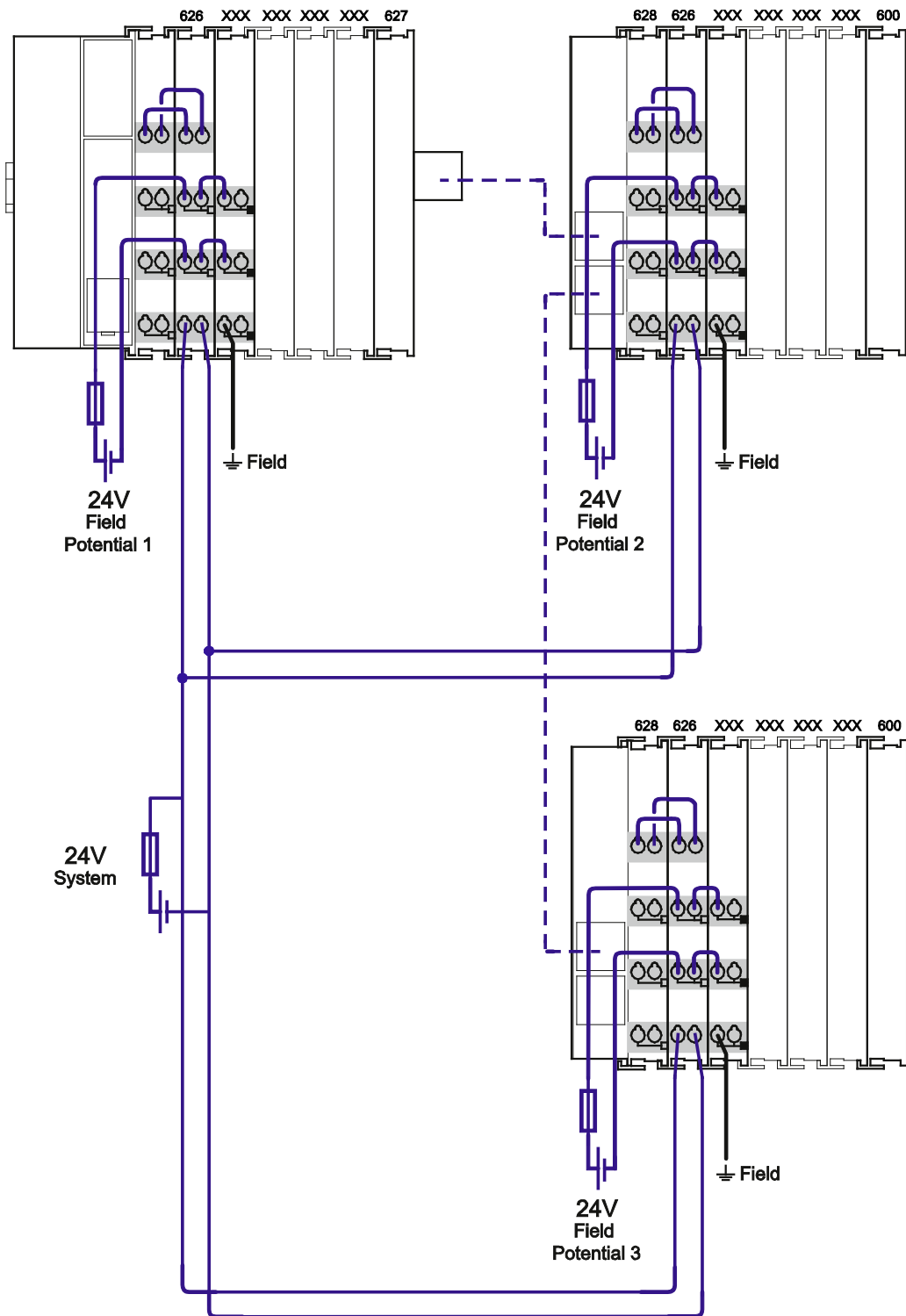


Figure 2: Isolation coupler module (example)

4.2 View

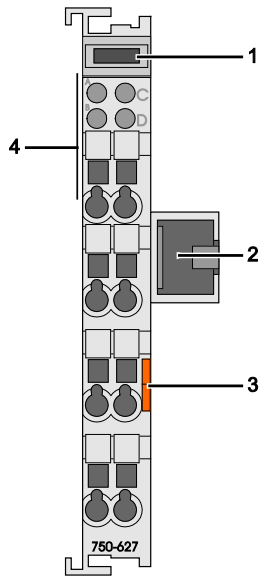


Figure 3: View

Table 3: Legend for the “View” figure

No.	Designation	Description	Details see chapter
1	---	Marking possibility with Mini-WSB	---
2	---	Bus connection, output (RJ-45 socket)	“Device Description” > “Connections”
3	---	Release clip	“Mounting” > “Insert and remove device”
4	---	Data contacts	“Device Description” > “Connections”

4.3 Connectors

4.3.1 Data Contacts/Internal Bus

The internal data bus is used for communication between fieldbus couplers/controllers and I/O modules, as well as the system power supply of the I/O modules. It consists of 6 data contacts that are available as self-cleaning gold contacts.

Correct data transmission via the internal data bus is ensured by the bus termination integrated in the I/O module.

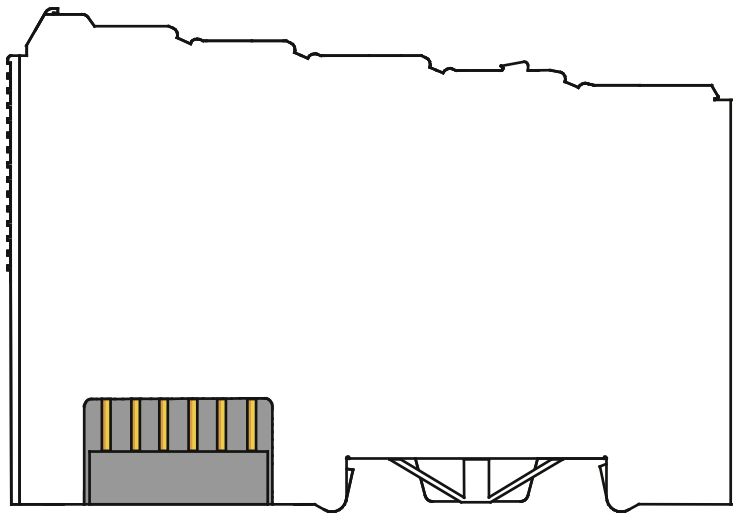


Figure 4: Data Contacts

NOTICE

Do not place the I/O modules on the gold spring contacts!

Do not place the I/O modules on the gold spring contacts in order to avoid soiling or scratching!

NOTICE



Ensure that the environment is well grounded!

The devices are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the devices, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, e.g. data contacts.

4.3.2 Power Jumper Contacts/Field Supply

The I/O module 750-627 has no power jumper contacts.

4.3.3 CAGE CLAMP® Connectors

The I/O module 750-627 has no CAGE CLAMP® connections.

4.3.4 RJ-45 Connectors

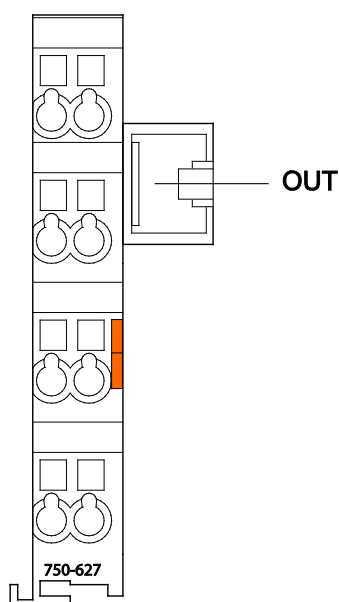


Figure 5: Connectors

Table 4: Legend for the “Connectors” figure

Designation	Connection	Function
OUT	RJ-45 without LED	Bus connection, output (RJ-45)

4.4 Display Elements

The I/O module 750-627 has no display elements.

4.5 Operating Elements

The I/O module 750-627 has no operating elements.

4.6 Technical Data

4.6.1 Device data

Table 5: Technical data – Device

Width	24 mm
Height (from upper edge of 35 DIN rail)	64 mm
Depth	100 mm
Weight	Approx. 65 g
Degree of protection	IP20

4.6.2 Supply

Table 6: Technical data – Supply

Power supply	Via internal bus (system voltage, 5 VDC)
Current via system voltage _{max.}	70 mA
Isolation	500 V system / supply

4.6.3 Connection type

Table 7: Technical data – Connection type interface

Number of interfaces	1
Bus connection	RJ-45 socket

Table 8: Technical data – Connection type module bus

Data contact	Slide contact, hard gold plated, self-cleaning
--------------	--

4.6.4 Communication

Table 9: Technical data – Communication

Number of coupler modules _{max.}	10
Number of partial nodes _{max.}	11
Distance between two components (end module and coupler module or coupler module and coupler module) _{max.}	5 m (10 m with capable copplers/controllers, see table in chapter “Module Bus Structure”)
Distance between all components (complete node) _{max.}	70 m
Transmission medium	Shielded copper wire (Ethernet patch cable) 4 x 2 x 0.25 mm ² , twisted pair, double shielding STP (Shielded Twisted-Pair)

4.6.5 Climatic Environmental Conditions

Table 10: Technical Data – Climatic Environmental Conditions

Operating temperature range	0 °C ... 55 °C
Storage temperature range	-25 °C ... +85 °C
Relative humidity without condensation	Max. 95 %
Resistance to harmful substances	Acc. to IEC 60068-2-42 and IEC 60068-2-43
Maximum pollutant concentration at relative humidity < 75 %	SO ₂ ≤ 25 ppm H ₂ S ≤ 10 ppm
Special conditions	Ensure that additional measures for components are taken, which are used in an environment involving: – dust, caustic vapors or gases – ionizing radiation

4.7 Approvals






Information


More information about approvals.

Detailed references to the approvals are listed in the document “Overview Approvals **WAGO-I/O-SYSTEM 750**”, which you can find via the internet under: www.wago.com > SERVICES > DOWNLOADS > Additional documentation and information on automation products > WAGO-I/O-SYSTEM 750 > System Description.



The following approvals have been granted to 750-627 I/O modules:

	Conformity Marking	
	cULus	UL508
	Korea Certification	MSIP-REM-W43-FSC750

The following Ex approvals have been granted to 750-627 I/O modules:

	DEKRA 11 ATEX 0203 X
	II 3 G Ex nA II T4

The following ship approvals have been granted to 750-627 I/O modules:

	GL (Germanischer Lloyd)	Cat. A, B, C, D (EMC 1)
	Federal Maritime and Hydrographic Agency	

4.8 Standards and Guidelines

750-627 I/O modules meet the following requirements on emission and immunity of interference:

EMC CE-Emission of interference	acc. to EN 61000-6-4: 2007
EMC CE-Immunity to interference	acc. to EN 61000-6-2: 2005
ATEX guideline	acc. to EN 60079-0: 2006 and acc. to EN 60079-15: 2005

5 Mounting

5.1 Mounting Sequence

Fieldbus couplers/controllers and I/O modules of the WAGO-I/O-SYSTEM 750 are snapped directly on a carrier rail in accordance with the European standard EN 50022 (DIN 35).

The reliable positioning and connection is made using a tongue and groove system. Due to the automatic locking, the individual devices are securely seated on the rail after installation.

Starting with the fieldbus coupler/controller, the I/O modules are mounted adjacent to each other according to the project design. Errors in the design of the node in terms of the potential groups (connection via the power contacts) are recognized, as the I/O modules with power contacts (blade contacts) cannot be linked to I/O modules with fewer power contacts.

NOTICE

Insert I/O modules only from the proper direction!

All I/O modules feature grooves for power jumper contacts on the right side. For some I/O modules, the grooves are closed on the top. Therefore, I/O modules featuring a power jumper contact on the left side cannot be snapped from the top. This mechanical coding helps to avoid configuration errors, which may destroy the I/O modules. Therefore, insert I/O modules only from the right and from the top.

Note



Don't forget the bus end module!

Always plug a bus end module 750-600 onto the end of the fieldbus node! You must always use a bus end module at all fieldbus nodes with WAGO-I/O-SYSTEM 750 fieldbus couplers/controllers to guarantee proper data transfer.

5.2 Inserting and Removing Devices

NOTICE

Perform work on devices only if they are de-energized!

Working on energized devices can damage them. Therefore, turn off the power supply before working on the devices.

5.2.1 Inserting the I/O Module

1. Position the I/O module so that the tongue and groove joints to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are engaged.

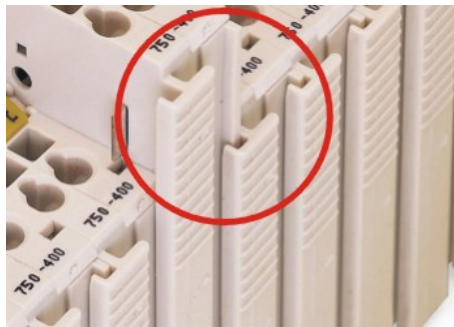


Figure 6: Insert I/O Module (example)

2. Press the I/O module into the assembly until the I/O module snaps into the carrier rail.

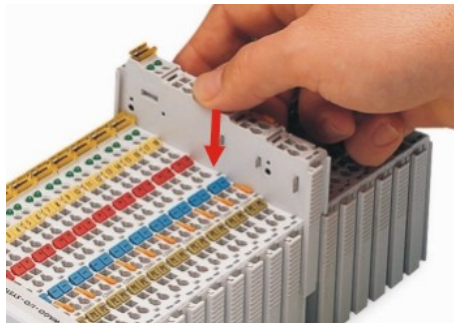


Figure 7: Snap the I/O Module into Place (example)

With the I/O module snapped in place, the electrical connections for the data contacts and power jumper contacts (if any) to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are established.

5.2.2 Removing the I/O Module

1. Remove the I/O module from the assembly by pulling the release tab.

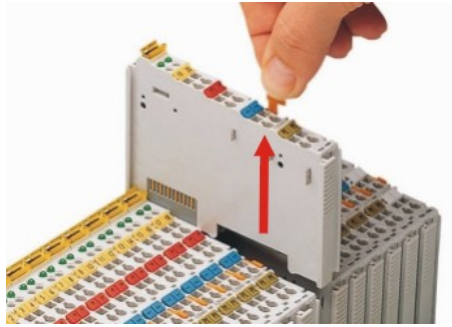


Figure 8: Removing the I/O Module (example)

6 Connect Devices

6.1 Connecting Cable to RJ-45 Socket

1. Connect the cable for bus extension to the RJ-45-socket (OUT) of the I/O module 750-627.
2. Connect the cable for bus extension to the top RJ-45-socket (IN) of the I/O module 750-628 of the next block.

Both blocks are connected to each other correctly.

7 Use in Hazardous Environments

The **WAGO-I/O-SYSTEM 750** (electrical equipment) is designed for use in Zone 2 hazardous areas.

The following sections include both the general identification of components (devices) and the installation regulations to be observed. The individual subsections of the “Installation Regulations” section must be taken into account if the I/O module has the required approval or is subject to the range of application of the ATEX directive.

7.1 Marking Configuration Examples

7.1.1 Marking for Europe According to ATEX and IEC-Ex

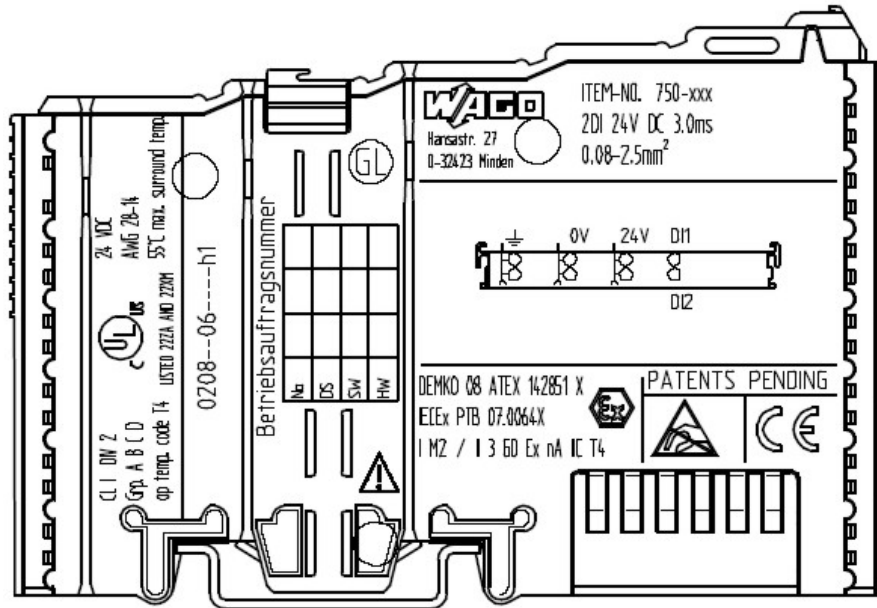


Figure 9: Side marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

DEMKO 08 ATEX 142851 X
IECEX PTB 07.0064X
I M2 / II 3 GD Ex nA IIC T4

Figure 10: Printing Text detail – Marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

Table 11: Description of marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

Printing on Text	Description
DEMKO 08 ATEX 142851 X IECEX PTB 07.0064X	Approval body and/or number of the examination certificate
I M2 / II 3 GD	Explosion protection group and Unit category
Ex nA	Type of ignition and extended identification
IIC	Explosion protection group
T4	Temperature class

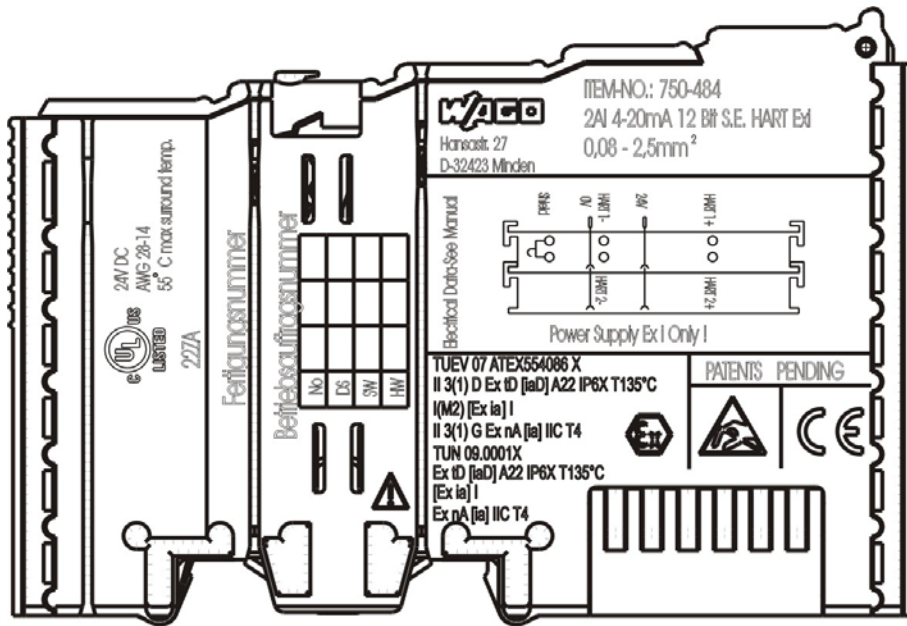


Figure 11: Side marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

TUEV 07 ATEX554086 X
II 3(1) D Ex tD [iaD] A22 IP6X T135°C
I(M2) [Ex ia] I
II 3(1) G Ex nA [ia] IIC T4
TUN 09.0001X
Ex tD [iaD] A22 IP6X T135°C
[Ex ia] I
Ex nA [ia] IIC T4



Figure 12: Text detail – Marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

Table 12: Description of marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

Inscription text	Description
TÜV 07 ATEX 554086 X TUN 09.0001X	Approving authority or certificate numbers
Dust	
II	Device group: All except mining
3(1)D	Device category: Zone 22 device (Zone 20 subunit)
Ex	Explosion protection mark
tD	Protection by enclosure
[iaD]	Approved in accordance with "Dust intrinsic safety" standard
A22	Surface temperature determined according to Procedure A, use in Zone 22
IP6X	Dust-tight (totally protected against dust)
T 135°C	Max. surface temp. of the enclosure (no dust bin)
Mining	
I	Device group: Mining
(M2)	Device category: High degree of safety
[Ex ia]	Explosion protection: Mark with category of type of protection intrinsic safety: Even safe when two errors occur
I	Device group: Mining
Gases	
II	Device group: All except mining
3(1)G	Device category: Zone 2 device (Zone 0 subunit)
Ex	Explosion protection mark
nA	Type of protection: Non-sparking operating equipment
[ia]	Category of type of protection intrinsic safety: Even safe when two errors occur
IIC	Explosion Group
T4	Temperature class: Max. surface temperature 135°C

7.1.2 Marking for America According to NEC 500

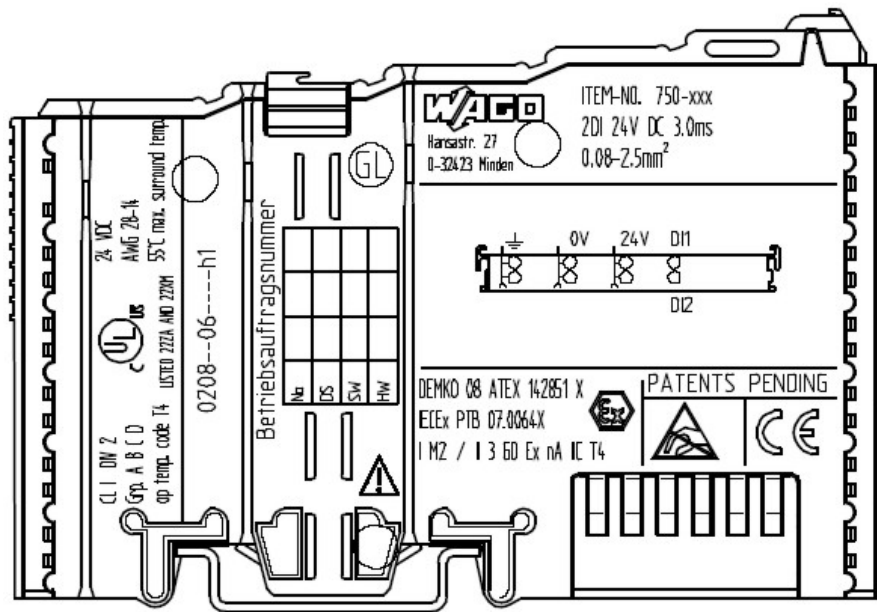


Figure 13: Side marking example for I/O modules according to NEC 500



Figure 14: Text detail – Marking example for I/O modules according to NEC 500

Table 13: Description of marking example for I/O modules according to NEC 500

Printing on Text	Description
CL 1	Explosion protection group (condition of use category)
DIV 2	Area of application (zone)
Grp. ABCD	Explosion group (gas group)
Optemp code T4	Temperature class

7.2 Installation Regulations

In the **Federal Republic of Germany**, various national regulations for the installation in explosive areas must be taken into consideration. The basis for this forms the working reliability regulation, which is the national conversion of the European guideline 99/92/E6. They are complemented by the installation regulation EN 60079-14. The following are excerpts from additional VDE regulations:

Table 14: VDE installation regulations in Germany

DIN VDE 0100	Installation in power plants with rated voltages up to 1000 V
DIN VDE 0101	Installation in power plants with rated voltages above 1 kV
DIN VDE 0800	Installation and operation in telecommunication plants including information processing equipment
DIN VDE 0185	lightning protection systems

The **USA** and **Canada** have their own regulations. The following are excerpts from these regulations:

Table 15: Installation regulations in USA and Canada

NFPA 70	National Electrical Code Art. 500 Hazardous Locations
ANSI/ISA-RP 12.6-1987	Recommended Practice
C22.1	Canadian Electrical Code

NOTICE

Notice the following points

When using the **WAGO-I/O SYSTEM 750** (electrical operation) with Ex approval, the following points are mandatory:

7.2.1 Special Conditions for Safe Operation of the ATEX and IEC Ex (acc. DEMKO 08 ATEX 142851X and IECEx PTB 07.0064)

The fieldbus-independent I/O modules of the WAGO-I/O-SYSTEM 750-.../...-... must be installed in an environment with degree of pollution 2 or better. In the final application, the I/O modules must be mounted in an enclosure with IP 54 degree of protection at a minimum with the following exceptions:

- I/O modules 750-440, 750-609 and 750-611 must be installed in an IP 64 minimum enclosure.
- I/O module 750-540 must be installed in an IP 64 minimum enclosure for 230 V AC applications.
- I/O module 750-440 may be used up to max. 120 V AC.

When used in the presence of combustible dust, all devices and the enclosure shall be fully tested and assessed in compliance with the requirements of IEC 61241-0:2004 and IEC 61241-1:2004.

When used in mining applications the equipment shall be installed in a suitable enclosure according to EN 60079-0:2006 and EN 60079-1:2007.

I/O modules fieldbus plugs or fuses may only be installed, added, removed or replaced when the system and field supply is switched off or the area exhibits no explosive atmosphere.

DIP switches, coding switches and potentiometers that are connected to the I/O module may only be operated if an explosive atmosphere can be ruled out.

I/O module 750-642 may only be used in conjunction with antenna 758-910 with a max. cable length of 2.5 m.

To exceed the rated voltage no more than 40%, the supply connections must have transient protection.

The permissible ambient temperature range is 0 °C to +55 °C.

7.2.2 Special conditions for safe use (ATEX Certificate TÜV 07 ATEX 554086 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the field bus independent I/O modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15, EN 61241-0 and EN 61241-1. For use as group I, electrical apparatus M2, the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.
2. If the interface circuits are operated without the field bus coupler station type 750-3../...-... (DEMKO 08 ATEX 142851 X), measures must be taken outside of the device so that the rating voltage is not being exceeded of more than 40% because of transient disturbances.
3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “CF-Card”, “USB”, “Fieldbus connection“, “Configuration and programming interface“, “antenna socket“, “D-Sub“ and the “Ethernet interface“. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.
6. For the type 750-601 the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The ambient temperature range is: $0^{\circ}\text{C} \leq T_a \leq +55^{\circ}\text{C}$ (for extended details please note certificate).

7.2.3 Special conditions for safe use (IEC-Ex Certificate TUN 09.0001 X)

1. For use as Dc- or Gc-apparatus (in zone 2 or 22) the fieldbus independent I/O modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15, IEC 61241-0 and IEC 61241-1. For use as group I, electrical apparatus M2, the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40% because of transient disturbances.
3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “CF-Card”, “USB”, “Fieldbus connection“, “Configuration and programming interface“, “antenna socket“, “D-Sub“ and the “Ethernet interface“. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.
6. For the type 750-601 the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The ambient temperature range is: $0^{\circ}\text{C} \leq T_a \leq +55^{\circ}\text{C}$ (For extensions please see the certificate).

7.2.4 Special conditions for safe use (ATEX Certificate DEKRA 11ATEX0203 X)

1. The components shall be installed in a suitable enclosure providing a degree of protection of at least IP54 according to EN 60529, taking into account the environmental conditions under which the equipment will be used.
2. When the temperature under rated conditions exceeds 70 °C at the cable or conduit entry point, or 80 °C at the branching point of the conductors, the temperature specification of the selected cable shall be in compliance with the actual measured temperature values.
3. Provisions shall be made to prevent the rated voltages from being exceeded by transient disturbances of more than 40 %.
4. Components may only be removed or inserted when the system supply and the field supply are switched off, or when the location is known to be non-hazardous.
5. Fieldbus connectors may only be disconnected or connected when the system supply is switched off, or when the location is known to be non-hazardous.
6. The fuses, present in de the system modules 750-6xx may only be replaced when the supply is switched off, or when the location is known to be non-hazardous.
7. Address selectors and ID switches may only be adjusted when the system supply is switched off, or when the location is known to be non-hazardous.
8. The ambient temperature range is: $0^{\circ}\text{C} \leq T_a \leq +55^{\circ}\text{C}$.

7.2.5 ANSI/ISA 12.12.01

This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only.

This equipment is to be fitted within tool-secured enclosures only.

WARNING

Explosion hazard!

Explosion hazard - substitution of components may impair suitability for Class I, Div. 2.

WARNING

Disconnect device when power is off and only in a non-hazardous area!

Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous near each operator accessible connector and fuse holder." When a fuse is provided, the following information shall be provided: "A switch suitable for the location where the equipment is installed shall be provided to remove the power from the fuse."

For devices with ETHERNET connectors:

"Only for use in LAN, not for connection to telecommunication circuits".

WARNING

Use only with antenna module 758-910!

Use Module 750-642 only with antenna module 758-910.

For Couplers/Controllers and Economy bus modules only: "The configuration Interface Service connector is for temporary connection only. Do not connect or disconnect unless the area is known to be nonhazardous. Connection or disconnection in an explosive atmosphere could result in an explosion.

WARNING

Devices containing fuses must not be fitted into circuits subject to over loads!

Devices containing fuses must not be fitted into circuits subject to over loads, e.g. motor circuits!

 **WARNING**

For devices equipped with SD card slots: Insert or remove the SD cards unless the area known to be free of ignitable concentrations of flammable gases or vapors!

Do not connect or disconnect SD-Card while circuit is live unless the area is known to be free of ignitable concentrations of flammable gases or vapors.

 **Information****Additional Information**

Proof of certification is available on request.

Also take note of the information given on the operating and assembly instructions.

The manual, containing these special conditions for safe use, must be readily available to the user.

List of Figures

Figure 1: Internal Data Bus Structure with Bus Extension (example)	13
Figure 2: Isolation coupler module (example)	16
Figure 3: View	17
Figure 4: Data Contacts	18
Figure 5: Connectors	19
Figure 6: Insert I/O Module (example)	24
Figure 7: Snap the I/O Module into Place (example).....	24
Figure 8: Removing the I/O Module (example).....	25
Figure 9: Side marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC	28
Figure 10: Printing Text detail – Marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC.....	28
Figure 11: Side marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC	29
Figure 12: Text detail – Marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC.....	29
Figure 13: Side marking example for I/O modules according to NEC 500	31
Figure 14: Text detail – Marking example for I/O modules according to NEC 500.....	31

List of Tables

Table 1: Number Notation.....	8
Table 2: Font Conventions	8
Table 3: Legend for the “View” figure	17
Table 4: Legend for the “Connectors” figure	19
Table 5: Technical data – Device	20
Table 6: Technical data – Supply	20
Table 7: Technical data – Connection type interface	20
Table 8: Technical data – Connection type module bus	20
Table 9: Technical data – Communication.....	20
Table 10: Technical Data – Climatic Environmental Conditions	21
Table 11: Description of marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC.....	28
Table 12: Description of marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC.....	30
Table 13: Description of marking example for I/O modules according to NEC 500.....	31
Table 14: VDE installation regulations in Germany	32
Table 15: Installation regulations in USA and Canada	32

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: +49/5 71/8 87 – 0
Fax: +49/5 71/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>

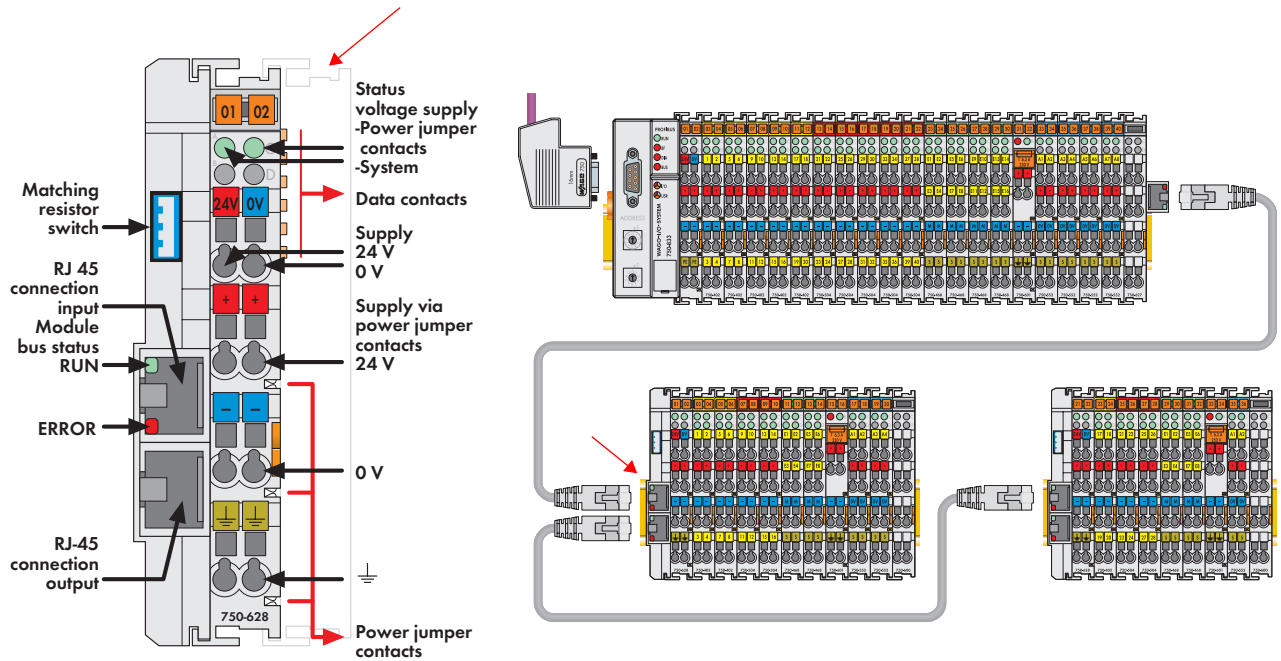


This Page Left Intentionally Blank

WAGO INTERNAL DATA BUS
EXTENSION COUPLER
MODULE

Operations & Maintenance Manual
December 2015

Internal Data Bus Extension Coupler Module



Delivered without miniature WSB markers

The coupler module for the internal data bus extension module 750-628 replaces the fieldbus coupler/controller at an I/O terminal block. It is also the mating piece for the end module 750-627. Plug the connecting cable into the top RJ-45 socket to establish the logical link to the fieldbus coupler/controller via end module 750-627. The extension is completely transparent for the fieldbus coupler/controller. All of the functions of the I/O module system are retained without any changes. A further extension to the system is provided by the bottom RJ-45 socket. This enables the entire system to be extended by 10 stages. The supply voltage for the field side and the internal electronics can be input separately. Both levels are electrically isolated from each other. Two diagnostic LEDs give information about the supply voltage for both the internal and field side. Two LEDs in the input socket indicate fault-free communication with the bus coupler. The extension module can be used as the last coupler module in the system (switch on matching resistor) or as a bridge between two I/O module assemblies.

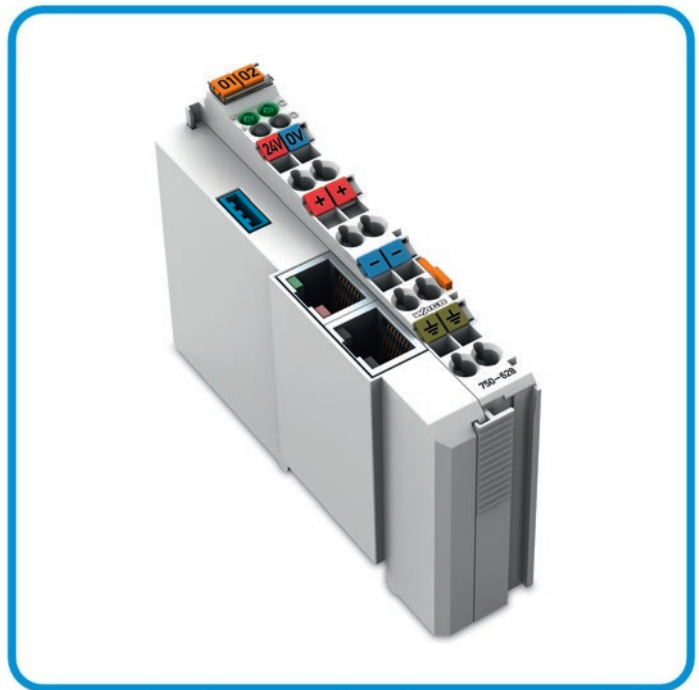
Installation note Attention:

To ensure safe, reliable operating states when using the internal data bus extension 750-627/-628 these states must be registered prior to startup with the following couplers or PLCs (refer to manual for supported couplers/PLCs). You must use the "WAGO Extension Setting" software for this (download: www.wago.com). Please note that only one terminating resistor may be activated in the whole system. Please complete the manufacturing number matrix on the right-hand side of the couplers when updating the firmware and internal operating parameters.

Description	Item No.	Pack. Unit
Internal Data Bus Extension Coupler Module	750-628	1
Accessories		
Miniature WSB Quick marking system		
plain	248-501	5
with marking	see pages 352 ... 353	
Approvals		
Also see "Approvals Overview" in Section 1		
Conformity marking	CE	
Shipbuilding	GL	
UL 508		

Technical Data	
Max. no. of I/O modules	64 (in the whole system)
Buscoupler connection	2 x RJ-45 socket (input + output)
Distance	5 m (10 m see manual), (end module and coupler or coupler and coupler)
Transmission medium	shielded copper wire (ETHERNET patch cable) 4 x 2 x 0.25 mm ² , twisted pair, double shielding
Power supply	24 V DC (-1.5 % ... +20 %)
Max. input current (24 V)	200 mA
Efficiency of the power supply	76 %
Inrush current	2.5 x continuous current
Internal current consumption (5 V)	150 mA
Total current for I/O modules (5 V)	400 mA
Voltage via power jumper contacts	24 V DC (-1.5 % ... +20 %)
Current via power jumper contacts (max.)	DC10 A
Isolation	500 V system/supply
Wire connection	CAGE CLAMP®
Cross sections	0.08 mm ² ... 2.5 mm ² / AWG 28 ... 14
Stripped lengths	8 ... 9 mm / 0.33 in
Width	25 mm
Weight	74.5 g
EMC: CE - immunity to interference	acc. to EN 61000-6-2 (2005)
EMC: CE - emission of interference	acc. to EN 61000-6-4 (2007)

This Page Left Intentionally Blank



WAGO-I/O-SYSTEM 750
Module Bus Extension Coupler Module
750-628

Version 1.1.0



© 2014 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

1	Notes about this Documentation.....	5
1.1	Validity of this Documentation.....	5
1.2	Copyright.....	5
1.3	Symbols.....	6
1.4	Number Notation.....	8
1.5	Font Conventions.....	8
2	Important Notes	9
2.1	Legal Bases	9
2.1.1	Subject to Changes	9
2.1.2	Personnel Qualifications.....	9
2.1.3	Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions	9
2.1.4	Technical Condition of Specified Devices.....	10
2.2	Safety Advice (Precautions).....	11
3	Internal Data Bus Structure with Bus Extension	13
4	Device Description	15
4.1	Power Supply	15
4.1.1	Isolation	15
4.2	View	17
4.3	Connectors.....	18
4.3.1	Data Contacts/Internal Bus.....	18
4.3.2	Power Jumper Contacts/Field Supply	19
4.3.3	CAGE CLAMP® Connectors	20
4.3.4	RJ-45 Connectors	21
4.4	Display Elements	22
4.5	Operating Elements	23
4.6	Schematic Diagram	23
4.7	Technical Data	24
4.7.1	Device data	24
4.7.2	Supply.....	24
4.7.3	Connection types	24
4.7.4	Communication	25
4.7.5	Climatic Environmental Conditions.....	25
4.8	Approvals	26
4.9	Standards and Guidelines	26
5	Mounting.....	27
5.1	Mounting Sequence.....	27
5.2	Inserting and Removing Devices	28
5.2.1	Inserting the I/O Module	28
5.2.2	Removing the I/O Module.....	29
6	Connect Devices	30
6.1	Connecting a Conductor to the CAGE CLAMP®	30
6.2	Connecting Cable to RJ-45 Socket	31
7	Commissioning	32

8	Use in Hazardous Environments	33
8.1	Marking Configuration Examples	34
8.1.1	Marking for Europe According to ATEX and IEC-Ex	34
8.1.2	Marking for America According to NEC 500	37
8.2	Installation Regulations	38
8.2.1	Special Conditions for Safe Operation of the ATEX and IEC Ex (acc. DEMKO 08 ATEX 142851X and IECEx PTB 07.0064)	39
8.2.2	Special conditions for safe use (ATEX Certificate TÜV 07 ATEX 554086 X)	40
8.2.3	Special conditions for safe use (IEC-Ex Certificate TUN 09.0001 X)	41
8.2.4	Special conditions for safe use (ATEX Certificate DEKRA 11ATEX0203 X)	42
8.2.5	ANSI/ISA 12.12.01	43
	List of Figures	45
	List of Tables	46

1 Notes about this Documentation



Note

Always retain this documentation!

This documentation is part of the product. Therefore, retain the documentation during the entire service life of the product. Pass on the documentation to any subsequent user. In addition, ensure that any supplement to this documentation is included, if necessary.

1.1 Validity of this Documentation

This documentation is only applicable to the I/O module 750-628 (Module Bus Extension Coupler Module).

The I/O module 750-628 shall only be installed and operated according to the instructions in this manual and in the manual for the used fieldbus coupler/controller.

NOTICE

Consider power layout of the WAGO-I/O-SYSTEM 750!

In addition to these operating instructions, you will also need the manual for the used fieldbus coupler/controller, which can be downloaded at www.wago.com. There, you can obtain important information including information on electrical isolation, system power and supply specifications.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.3 Symbols

 **DANGER****Personal Injury!**

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **DANGER****Personal Injury Caused by Electric Current!**

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 **WARNING****Personal Injury!**

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

 **CAUTION****Personal Injury!**

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE**Damage to Property!**

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE**Damage to Property Caused by Electrostatic Discharge (ESD)!**

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note**Important Note!**

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.



Information

Additional Information:

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.4 Number Notation

Table 1: Number Notation

Number code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.5 Font Conventions

Table 2: Font Conventions

Font type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Programme\WAGO-I/O-CHECK</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualifications

All sequences implemented on WAGO-I/O-SYSTEM 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

All changes to the coupler or controller should always be carried out by qualified personnel with sufficient skills in PLC programming.

2.1.3 Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions

Fieldbus couplers, fieldbus controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to actuators or higher-level control systems. Using programmable controllers, the signals can also be (pre-) processed.

The devices have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the devices in wet and dusty environments is prohibited.

Operating the WAGO-I/O-SYSTEM 750 devices in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section “Device Description” > “Standards and Guidelines” in the manual for the used fieldbus coupler/controller.

Appropriate housing (per 94/9/EG) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

2.1.4 Technical Condition of Specified Devices

The devices to be supplied ex works are equipped with hardware and software configurations, which meet the individual application requirements. WAGO Kontakttechnik GmbH & Co. KG will be exempted from any liability in case of changes in hardware or software as well as to non-compliant usage of devices.

Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

2.2 Safety Advice (Precautions)

For installing and operating purposes of the relevant device to your system the following safety precautions shall be observed:



DANGER

Do not work on devices while energized!

All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.

DANGER

Install the device only in appropriate housings, cabinets or in electrical operation rooms!

The WAGO-I/O-SYSTEM 750 and its components are an open system. As such, install the system and its components exclusively in appropriate housings, cabinets or in electrical operation rooms. Allow access to such equipment and fixtures to authorized, qualified staff only by means of specific keys or tools.

NOTICE

Replace defective or damaged devices!

Replace defective or damaged device/module (e.g., in the event of deformed contacts), since the long-term functionality of device/module involved can no longer be ensured.

NOTICE

Protect the components against materials having seeping and insulating properties!

The components are not resistant to materials having seeping and insulating properties such as: aerosols, silicones and triglycerides (found in some hand creams). If you cannot exclude that such materials will appear in the component environment, then install the components in an enclosure being resistant to the above-mentioned materials. Clean tools and materials are imperative for handling devices/modules.

NOTICE

Clean only with permitted materials!

Clean soiled contacts using oil-free compressed air or with ethyl alcohol and leather cloths.

NOTICE**Do not use any contact spray!**

Do not use any contact spray. The spray may impair contact area functionality in connection with contamination.

NOTICE**Do not reverse the polarity of connection lines!**

Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

**NOTICE****Avoid electrostatic discharge!**

The devices are equipped with electronic components that may be destroyed by electrostatic discharge when touched. Please observe the safety precautions against electrostatic discharge per DIN EN 61340-5-1/-3. When handling the devices, please ensure that environmental factors (personnel, work space and packaging) are properly grounded.

NOTICE**Feed system power supply in parallel!**

The flow of equalizing currents within the bus electronics can destroy the devices. Therefore, connect the system power supply to all supply points of the fieldbus node in parallel. This prevents potential differences and equalizing currents from forming.

NOTICE**Perform insulation tests with direct current (DC)!**

Both the supply voltage and control voltage side are capacitively coupled to the DIN rail. If the modules are mounted on the DIN rail, application of an AC voltage between the two potentials can lead to the destruction of the device.

Use only direct current (DC) for insulation testing.

To avoid destroying the device, discharge the device completely before applying the test voltage again.

3 Internal Data Bus Structure with Bus Extension

With an I/O module 750-627 (end module for bus extension) and one or more I/O modules 750-628 (coupler module for bus extension), a fieldbus node can be divided into max. 11 I/O module blocks (hereinafter referred to as "blocks"). The divided fieldbus nodes still act as a logical unit for the fieldbus coupler/controller. All of the functions of the I/O module system are retained without any changes.

The first block is terminated with an I/O module 750-627 (end module for bus extension). Each subsequent block is started with an I/O module 750-628 (coupler module for bus extension) instead of a fieldbus coupler/controller and terminated with an I/O module 750-600 (end module).

Note



Number of I/O modules independent of bus extension!

The number of total I/O modules permitted in the fieldbus node does not increase by the use of an I/O module for bus extension.

Use an ETHERNET patch cable as a connection cable with the following properties between two blocks:

- Category 5
- Double-insulated shielded twisted pair (STP)
- Cooper
- $4 \times 2 \times 0.25 \text{ mm}^2$

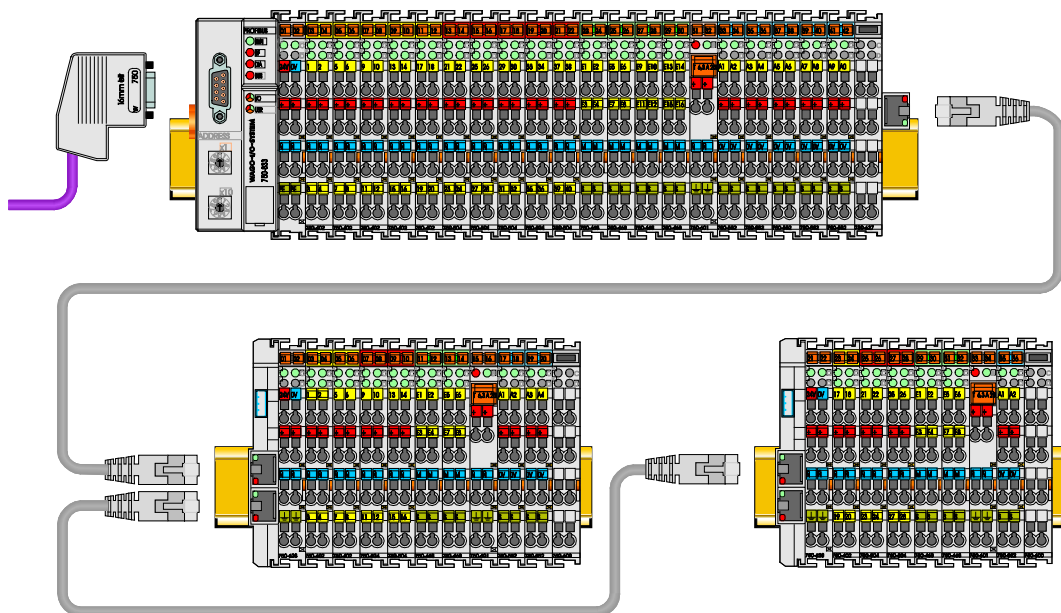


Figure 1: Internal Data Bus Structure with Bus Extension (example)

For the devices as delivered, the cable length between two sub-nodes can be max. 5 meters. Many fieldbus couplers/controllers can be configured, so that the permitted cable length can be increased to max. 10 meters. This setting can be made using the "WAGO Extension Setting Tool" (759-314), which can be downloaded at www.wago.com at no cost.

Note

Note the maximum total cable length of 70 meters!

The length of all connection cables cannot exceed 70 meters.

The maximum cable length permitted cannot be increased for the following fieldbus couplers/controllers:

Item No.	Name
750-300	II/O-LIGHTBUS Fieldbus Coupler
750-301	PROFIBUS Fieldbus Coupler
750-304	INTERBUS Fieldbus Coupler
750-305	CAL Fieldbus Coupler
750-309	LON Fieldbus Coupler
750-310	CC-LINK Fieldbus Coupler
750-312	MODBUS Fieldbus Coupler, RS-485
750-313	SDS Fieldbus Coupler
750-314	MODBUS Fieldbus Coupler, RS-232
750-315	MODBUS Fieldbus Coupler, RS-485
750-316	MODBUS Fieldbus Coupler, RS-232
750-327	CANopen Fieldbus Coupler
750-331	PROFIBUS Fieldbus Coupler
750-339	FireWire Fieldbus Coupler
750-804	INTERBUS Fieldbus Controller
750-812	MODBUS Fieldbus Controller, RS-485
750-814	MODBUS Fieldbus Controller, RS-232
750-815	MODBUS Fieldbus Controller, RS-485
750-816	MODBUS Fieldbus Controller, RS-232

The above table does not include all versions of the fieldbus couplers/controllers. In contrast to standard items, some specific variants may indeed support a configuration change. Corresponding information is given in the documentation for the specific fieldbus coupler/controller.

4 Device Description

The I/O module 750-628 (coupler module for bus extension) replaces the fieldbus coupler/controller at a block. More information on the basic structure of a fieldbus node is available in Section "Internal Data Bus Structure with Bus Extension".

The I/O module has a switch with which a terminating resistor can be connected. Use in the last, as well as in a block in the middle of a fieldbus node is possible. To connect with the blocks, the I/O module has two RJ-45 sockets.

Two diagnostic LEDs indicate the state of the system and field supply voltage for the relevant block. The two LEDs of an RJ-45 socket indicate error-free communication with the adjacent blocks. The supply voltage for the field side and the internal system can be input separately. Both levels are electrically isolated from each other.

NOTICE

Do not exceed maximum current via power jumper contacts!

The maximum current to flow through the power jumper contacts is 10 A. Greater currents can damage the contacts.

When configuring your system, ensure that this current is not exceeded. If exceeded, insert an additional supply module.

4.1 Power Supply

4.1.1 Isolation

Within the fieldbus node, there are three electrically isolated potentials:

- Electrically isolated fieldbus interface via transformer
- Electronics of the fieldbus couplers/controllers and the I/O modules (internal bus)
- All I/O modules have an electrical isolation between the electronics (internal bus, logic) and the field electronics. Some digital and analog input modules have each channel electrically isolated, please see catalog.

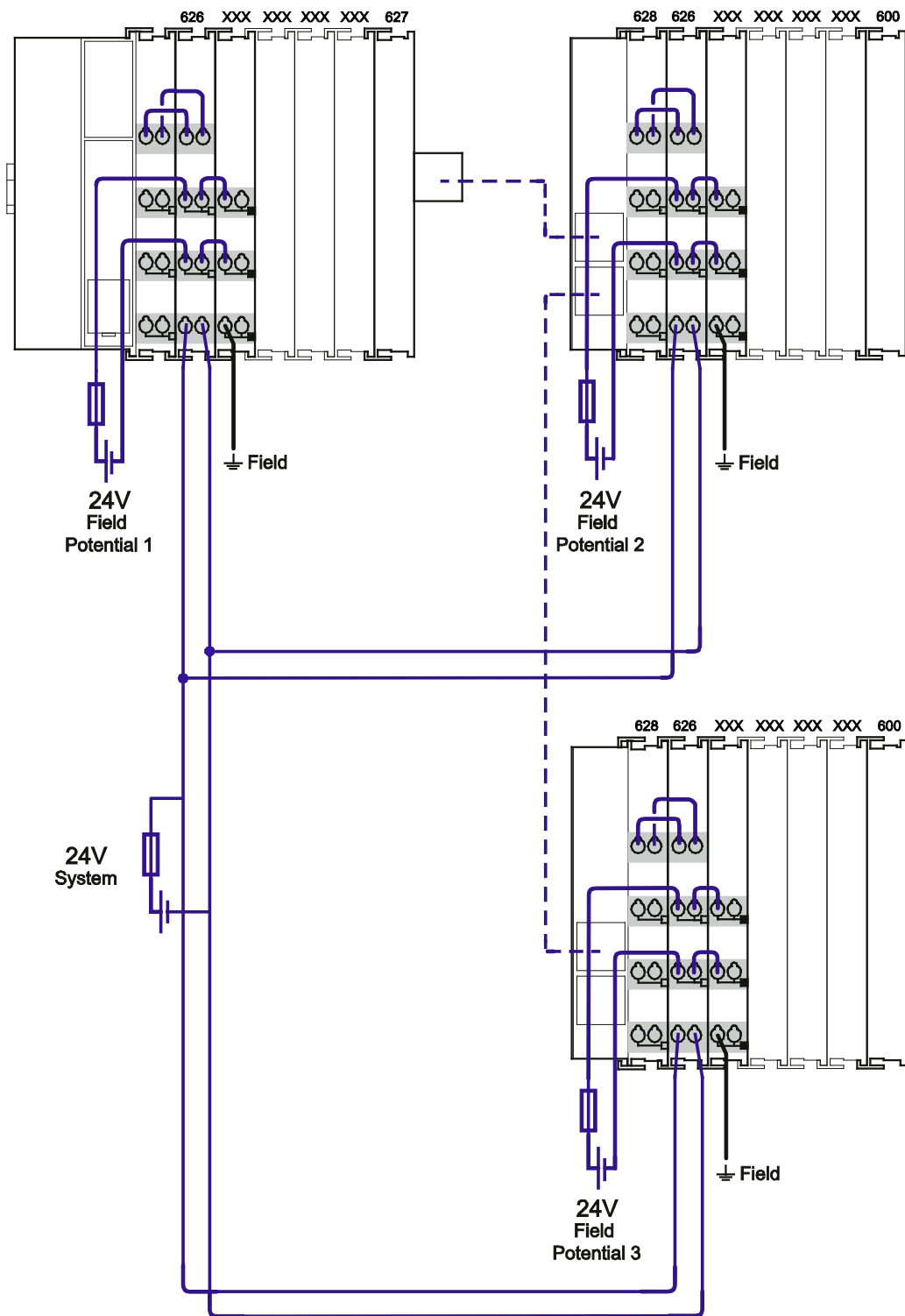


Figure 2: Isolation coupler module (example)

4.2 View

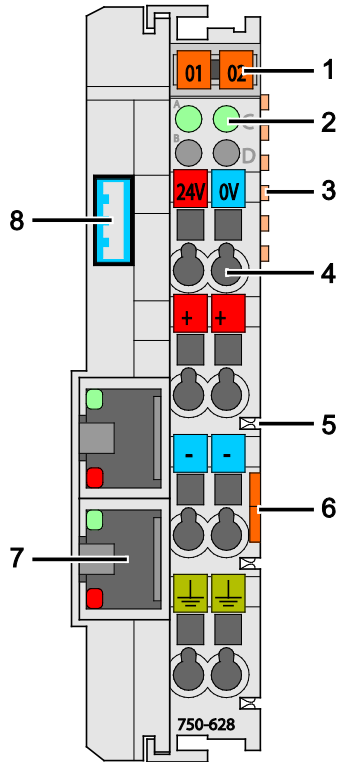


Figure 3: View

Table 3: Legend for the “View” figure

No.	Designation	Description	Details see chapter
1	---	Marking possibility with Mini-WSB	---
2	A, B	Status-LED	“Device Description” > “Display Elements”
3	---	Data contacts	“Device Description” > “Connectors”
4	1 ... 8	CAGE CLAMP®-connectors	“Device Description” > “Connectors”
5	---	Power jumper contact	“Device Description” > “Connectors”
6	---	Release clip	“Mounting” > “Insert and remove device”
7	---	Bus connection (RJ-45)	<ul style="list-style-type: none"> • “Device Description” > “Display Elements” • “Device Description” > “Connectors”
8	---	Terminating resistor	“Device Description” > “Operating Elements”

4.3 Connectors

4.3.1 Data Contacts/Internal Bus

Communication between the fieldbus coupler/controller and the I/O modules as well as the system supply of the I/O modules is carried out via the internal bus. It is comprised of 6 data contacts, which are available as self-cleaning gold spring contacts.

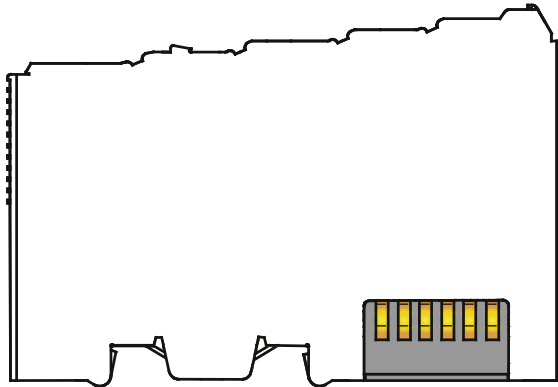


Figure 4: Data Contacts

NOTICE

Do not place the I/O modules on the gold spring contacts!

Do not place the I/O modules on the gold spring contacts in order to avoid soiling or scratching!

NOTICE



Ensure that the environment is well grounded!

The devices are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the devices, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, e.g. data contacts.

4.3.2 Power Jumper Contacts/Field Supply

The Module Bus Extension Coupler Module 750-628 is equipped with 3 self-cleaning power contacts for transferring of the field-side power supply to down-circuit I/O modules. These contacts are designed as spring contacts.

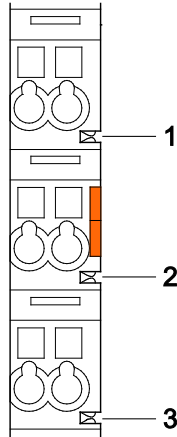


Figure 5: Power Jumper Contacts

Table 4: Legend for the “Power Jumper Contacts” Figure

Contact	Type	Function
1	Spring contact	Potential transmission (U_V) for field supply
2	Spring contact	Potential transmission (0 V) for field supply
3	Spring contact	Potential transmission (ground) for field supply

NOTICE

Do not exceed maximum current via power jumper contacts!

The maximum current to flow through the power jumper contacts is 10 A. Greater currents can damage the contacts.

When configuring your system, ensure that this current is not exceeded. If exceeded, insert an additional supply module.

4.3.3 CAGE CLAMP® Connectors

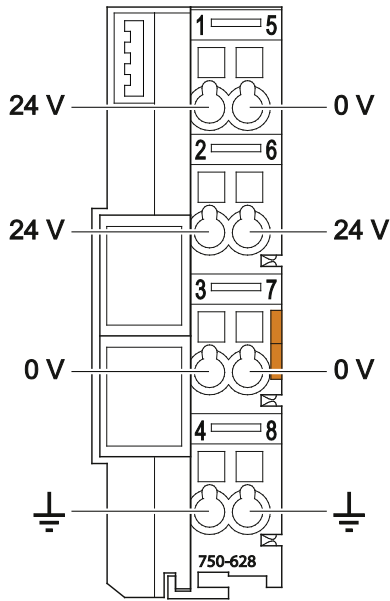


Figure 6: CAGE CLAMP® Connectors

Table 5: Legend for the „CAGE CLAMP®-Connectors“ Figure

Designation	Connection	Function
24 V	1	System supply voltage: 24 VDC
0 V	5	System supply voltage: 0 V
24 V	2	Field supply voltage: 24 V
	6	Field supply voltage: 24 V
0 V	3	Field supply voltage: 0 V
	7	Field supply voltage: 0 V
Ground	4	Field supply voltage: ground
	8	Field supply voltage: ground

4.3.4 RJ-45 Connectors

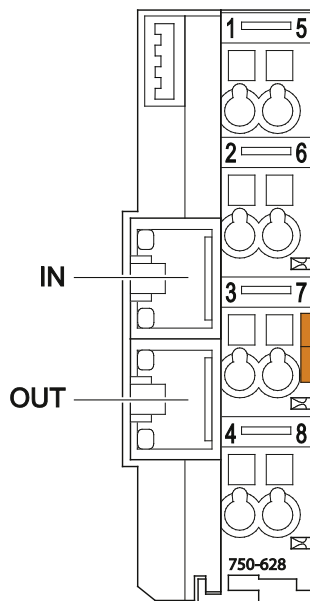


Figure 7: Connectors

Table 6: Legend to the „Connectors“ Figure

Designation	Connection	Function
IN	RJ-45 with LED	Bus connection, input (RJ-45)
OUT	RJ-45 with LED	Bus connection, output (RJ-45)

4.4 Display Elements

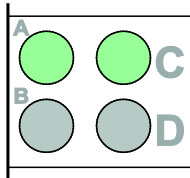


Figure 8: Display Elements – Power supply

Table 7: Legend for „Display Elements – Power supply“

Designation	LED	State	Function
Status of operating voltage – system	A	Off	No 24 VDC system supply voltage.
		Green	24 VDC system supply voltage.
Status of operating voltage – power jumper contacts	C	Off	No 24 VDC voltage supply via power jumper contacts.
		Green	24 VDC voltage supply via power jumper contacts.

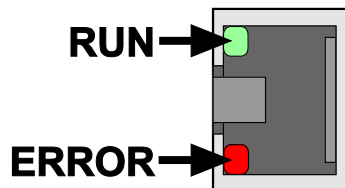


Figure 9: Display Elements – Bus connection

Table 8: Legend for „Display Elements – Bus connection“

Designation	LED	State	Function
RUN	Green	Off	No Communication
		On	Communication is available
ERROR	Red	Off	Communication is available
		On	No Communication

4.5 Operating Elements

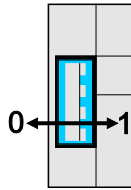


Figure 10: Operating Elements

Table 9: Legend for the „Operating Elements“ figure

Designation	State	Function
Terminating resistor	0	Terminating resistor is switch off, I/O module is not last coupler module
	1	Terminating resistor is switch on, I/O module is last coupler module

4.6 Schematic Diagram

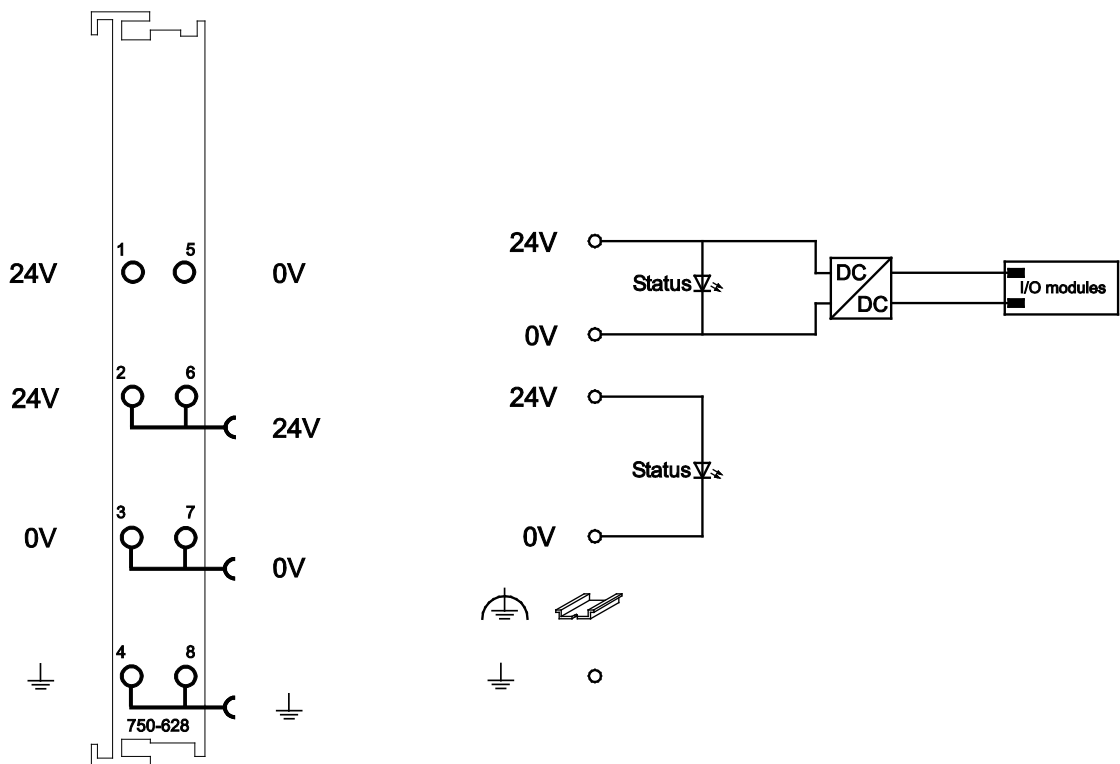


Figure 11: Schematic diagram

4.7 Technical Data

4.7.1 Device data

Table 10: Technical Data – Device

Width	25 mm
Height (from upper edge of 35 DIN rail)	64 mm
Depth	100 mm
Weight	Approx. 70 g
Degree of protection	IP20

4.7.2 Supply

Table 11: Technical Data – Supply

System supply	
Input voltage supply	24 VDC (-15 % ... +20 %)
Input current consumption _{max.}	200 mA
Input inrush current	2,5 x continuous current
Voltage output	5 VDC
Current output _{max.}	400 mA
Field supply	
Voltage	24 VDC (-25 % ... +30 %)
Current consumption _{max.}	10 A
Current output _{max.}	10 A
Isolation	
System/supply	500 V

4.7.3 Connection types

Table 12: Technical Data – Connection types: interface

Number of interfaces	2
Bus connection	RJ-45 socket (input and output)

Table 13: Technical Data – Connection types: field wiring

Wire connection	CAGE CLAMP®
Cross section	0,25 mm ² ... 2,5 mm ² / AWG 24 ... 14
Stripped lengths	8 mm ... 9 mm / 0,33 in

Table 14: Technical Data – Connection types: power jumper contact

Power jumper contact	Spring contacts, self-cleaning
Voltage drop at I_{max} .	< 1 V at 64 I/O modules

Table 15: Technical Data – Connection types: internal data bus

Data contacts	Slide contacts, self-cleaning
---------------	-------------------------------

4.7.4 Communication

Table 16: Technical Data – Communication

Number of I/O modules of each node max.	Depending on the used coupler/controller
Number of I/O modules of each coupler module max.	Hardware/software version [item number] < XXXX0103... : 40
	Hardware/software version [item number] >= XXXX0103... : 64
Number of coupler modules max.	10
Number of partial nodes max.	11
Distance between two components (end module and coupler module or coupler module and coupler module) max.	5 m (10 m with capable copplers/controllers, see table in chapter “Module Bus Structure”)
Distance between all components (complete node) max.	70 m
Transmission medium	Shielded copper wire (Ethernet patch cable) 4 x 2 x 0.25 mm ² , twisted pair, double shielding STP (Shielded Twisted-Pair)

4.7.5 Climatic Environmental Conditions

Table 17: Technical Data – Climatic Environmental Conditions

Operating temperature range	0 °C ... 55 °C
Storage temperature range	-25 °C ... +85 °C
Relative humidity without condensation	Max. 95 %
Resistance to harmful substances	Acc. to IEC 60068-2-42 and IEC 60068-2-43
Maximum pollutant concentration at relative humidity < 75 %	SO ₂ ≤ 25 ppm H ₂ S ≤ 10 ppm
Special conditions	Ensure that additional measures for components are taken, which are used in an environment involving: – dust, caustic vapors or gases – ionizing radiation

4.8 Approvals



Information

More information about approvals.

Detailed references to the approvals are listed in the document “Overview Approvals **WAGO-I/O-SYSTEM 750**”, which you can find via the internet under: www.wago.com > SERVICES > DOWNLOADS > Additional documentation and information on automation products > WAGO-I/O-SYSTEM 750 > System Description.

The following approvals have been granted to 750-628 I/O modules:



Conformity Marking



cULus UL508



Korea Certification MSIP-REM-W43-IDE750

The following Ex approvals have been granted to 750-628 I/O modules:



DEKRA 11 ATEX 0203 X

II 3 G Ex nA II T4

The following ship approvals have been granted to 750-628 I/O modules:



GL (Germanischer Lloyd) Cat. A, B, C, D (EMC 1)



Federal Maritime and Hydrographic Agency

4.9 Standards and Guidelines

All variations of 750-628 I/O modules meet the following requirements on emission and immunity of interference:

EMC CE-Emission of interference	acc. to EN 61000-6-4: 2007
EMC CE-Immunity to interference	acc. to EN 61000-6-2: 2005
ATEX guideline	acc. to EN 60079-0: 2006 and acc. to EN 60079-15: 2005

5 Mounting

5.1 Mounting Sequence

Fieldbus couplers/controllers and I/O modules of the WAGO-I/O-SYSTEM 750 are snapped directly on a carrier rail in accordance with the European standard EN 50022 (DIN 35).

The reliable positioning and connection is made using a tongue and groove system. Due to the automatic locking, the individual devices are securely seated on the rail after installation.

Starting with the fieldbus coupler/controller, the I/O modules are mounted adjacent to each other according to the project design. Errors in the design of the node in terms of the potential groups (connection via the power contacts) are recognized, as the I/O modules with power contacts (blade contacts) cannot be linked to I/O modules with fewer power contacts.

CAUTION

Risk of injury due to sharp-edged blade contacts!

The blade contacts are sharp-edged. Handle the I/O module carefully to prevent injury.

NOTICE

Insert I/O modules only from the proper direction!

All I/O modules feature grooves for power jumper contacts on the right side. For some I/O modules, the grooves are closed on the top. Therefore, I/O modules featuring a power jumper contact on the left side cannot be snapped from the top. This mechanical coding helps to avoid configuration errors, which may destroy the I/O modules. Therefore, insert I/O modules only from the right and from the top.

Note



Don't forget the bus end module!

Always plug a bus end module 750-600 onto the end of the fieldbus node! You must always use a bus end module at all fieldbus nodes with WAGO-I/O-SYSTEM 750 fieldbus couplers/controllers to guarantee proper data transfer.

5.2 Inserting and Removing Devices

NOTICE

Perform work on devices only if they are de-energized!

Working on energized devices can damage them. Therefore, turn off the power supply before working on the devices.

5.2.1 Inserting the I/O Module

1. Position the I/O module so that the tongue and groove joints to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are engaged.

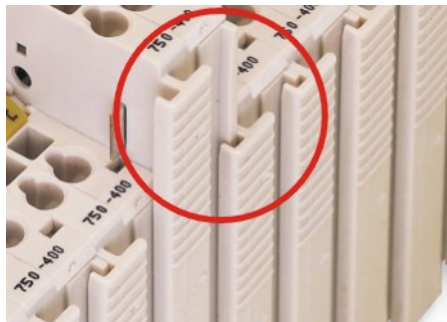


Figure 12: Insert I/O Module (Example)

2. Press the I/O module into the assembly until the I/O module snaps into the carrier rail.

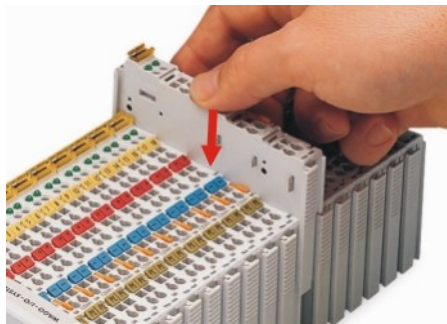


Figure 13: Snap the I/O Module into Place (Example)

With the I/O module snapped in place, the electrical connections for the data contacts and power jumper contacts (if any) to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are established.

5.2.2 Removing the I/O Module

1. Remove the I/O module from the assembly by pulling the release tab.

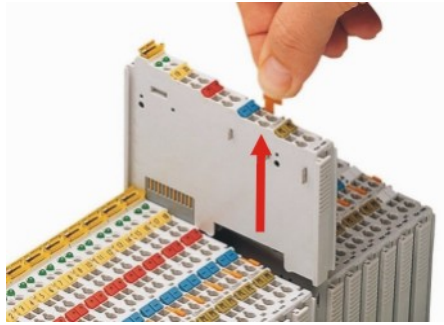


Figure 14: Removing the I/O Module (Example)

Electrical connections for data or power jumper contacts are disconnected when removing the I/O module.

6 Connect Devices

NOTICE

Feed system power supply in parallel!

The flow of equalizing currents within the bus electronics can destroy the devices. Therefore, connect the system power supply to all supply points of the fieldbus node in parallel. This prevents potential differences and equalizing currents from forming.

6.1 Connecting a Conductor to the CAGE CLAMP®

The WAGO CAGE CLAMP® connection is appropriate for solid, stranded and finely stranded conductors.

Note



Only connect one conductor to each CAGE CLAMP®!

Only one conductor may be connected to each CAGE CLAMP®.

Do not connect more than one conductor at one single connection!

If more than one conductor must be routed to one connection, these must be connected in an up-circuit wiring assembly, for example using WAGO feed-through terminals.

Exception:

If it is unavoidable to jointly connect 2 conductors, then you must use a ferrule to join the wires together. The following ferrules can be used:

Length:	8 mm
Nominal cross section _{max.} :	1 mm ² for 2 conductors with 0.5 mm ² each
WAGO product:	216-103 or products with comparable properties

1. For opening the CAGE CLAMP® insert the actuating tool into the opening above the connection.
2. Insert the conductor into the corresponding connection opening.
3. For closing the CAGE CLAMP® simply remove the tool. The conductor is now clamped firmly in place.

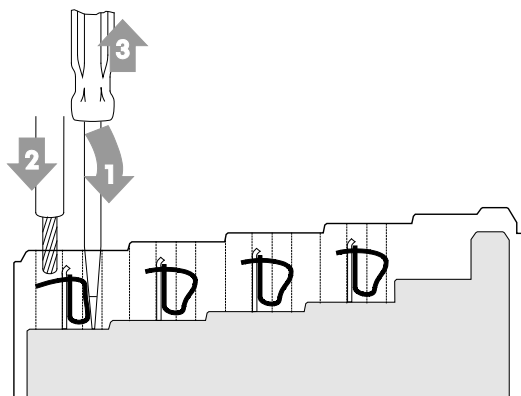


Figure 15: Connecting a Conductor to a CAGE CLAMP®

6.2 Connecting Cable to RJ-45 Socket

1. Connect the cable for bus extension from the previous block to the top RJ-45 socket (IN) of the I/O module.

To connect the block to another block:

2. Connect the cable for bus extension of the block currently being considered to the bottom RJ-45 socket (OUT) of the I/O module.
3. Connect the cable for bus extension to the top RJ-45 socket (IN) of the I/O module of the next block.

Note



Connection to the I/O module!

Up to 40 I/O modules can be connected to an I/O module 750-628 with hardware/software version [series item number] < XXXX-0103 and up to 64 I/O modules can be connected to an I/O module 750-628 with hardware/software version [series item number] ≥ XXXX-0103.

7 Commissioning

DANGER

Caution in case of failure of the I/O module!

If the I/O module 750-628 fails, communication is interrupted to the subsequent I/O module blocks. Make sure that people or equipment are not placed at risk if the I/O module 750-628 fails.

DANGER

Switch on the terminating resistor at the last I/O module 750-628 only!

Only switch on the terminating resistor at the I/O module 750-628 located in the last of the I/O module blocks of the split fieldbus node. If the terminating resistor of another I/O module 750-628 in the fieldbus node is activated, faulty functions of the fieldbus node may occur (e.g., accidental connection of outputs).

If the current block represents the last block of the fieldbus node:

1. Switch on the terminating resistor.
2. Flip the switch to the right.

If the current block is connected to another block, i.e., not the last block in the fieldbus node:

1. Switch off the terminating resistor.
2. Flip the switch to the left.

The terminating resistor is activated.

8 Use in Hazardous Environments

The **WAGO-I/O-SYSTEM 750** (electrical equipment) is designed for use in Zone 2 hazardous areas.

The following sections include both the general identification of components (devices) and the installation regulations to be observed. The individual subsections of the “Installation Regulations” section must be taken into account if the I/O module has the required approval or is subject to the range of application of the ATEX directive.

8.1 Marking Configuration Examples

8.1.1 Marking for Europe According to ATEX and IEC-Ex

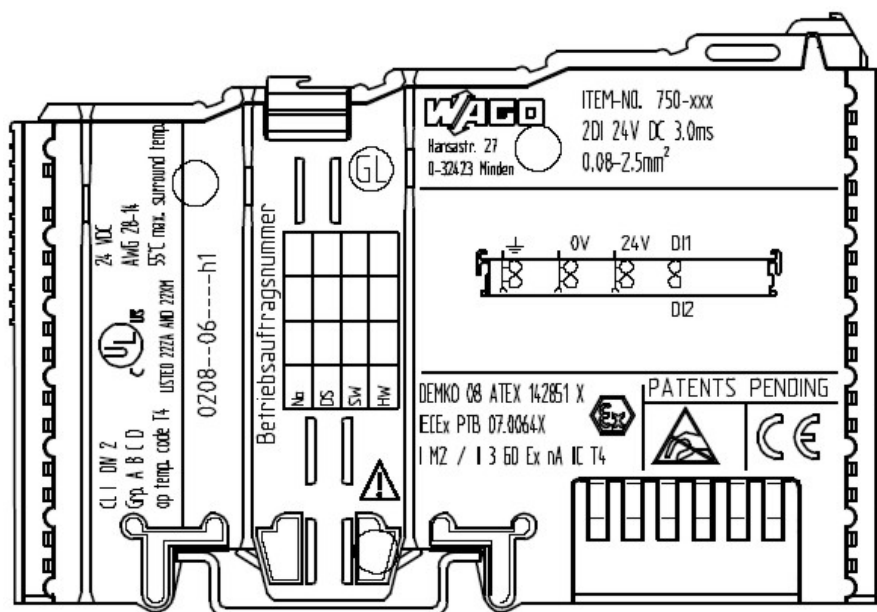


Figure 16: Side marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

DEMKO 08 ATEX 142851 X
IECEX PTB 07.0064X
I M2 / II 3 GD Ex nA IIC T4



Figure 17: Printing Text detail – Marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

Table 18: Description of marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

Printing on Text	Description
DEMKO 08 ATEX 142851 X IECEX PTB 07.0064X	Approval body and/or number of the examination certificate
I M2 / II 3 GD	Explosion protection group and Unit category
Ex nA	Type of ignition and extended identification
IIC	Explosion protection group
T4	Temperature class

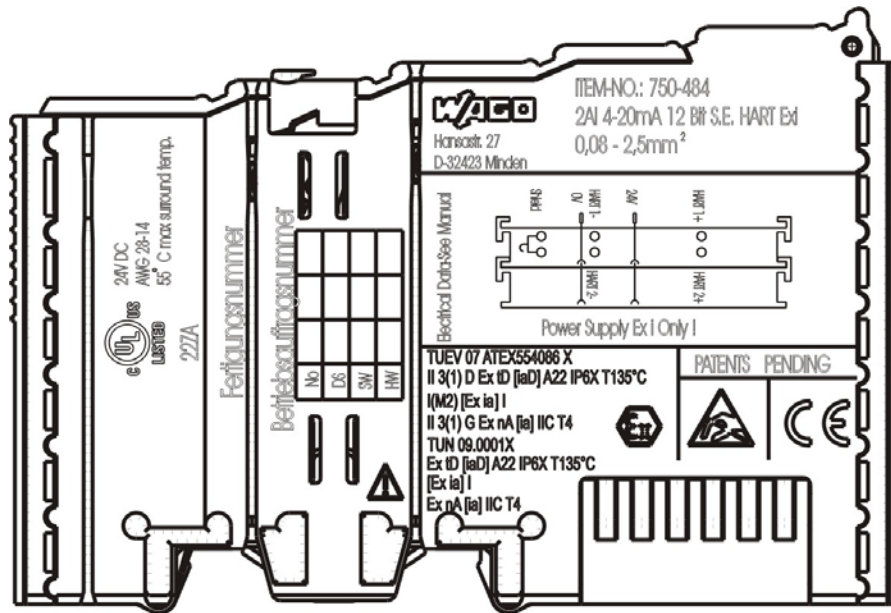


Figure 18: Side marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

TUEV 07 ATEX554086 X
II 3(1) D Ex tD [iaD] A22 IP6X T135°C
I(M2) [Ex ia] I
II 3(1) G Ex nA [ia] IIC T4
TUN 09.0001X
Ex tD [iaD] A22 IP6X T135°C
[Ex ia] I
Ex nA [ia] IIC T4



Figure 19: Text detail – Marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

Table 19: Description of marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

Inscription text	Description
TÜV 07 ATEX 554086 X TUN 09.0001X	Approving authority or certificate numbers
Dust	
II	Device group: All except mining
3(1)D	Device category: Zone 22 device (Zone 20 subunit)
Ex	Explosion protection mark
tD	Protection by enclosure
[iaD]	Approved in accordance with "Dust intrinsic safety" standard
A22	Surface temperature determined according to Procedure A, use in Zone 22
IP6X	Dust-tight (totally protected against dust)
T 135°C	Max. surface temp. of the enclosure (no dust bin)
Mining	
I	Device group: Mining
(M2)	Device category: High degree of safety
[Ex ia]	Explosion protection: Mark with category of type of protection intrinsic safety: Even safe when two errors occur
I	Device group: Mining
Gases	
II	Device group: All except mining
3(1)G	Device category: Zone 2 device (Zone 0 subunit)
Ex	Explosion protection mark
nA	Type of protection: Non-sparking operating equipment
[ia]	Category of type of protection intrinsic safety: Even safe when two errors occur
IIC	Explosion Group
T4	Temperature class: Max. surface temperature 135°C

8.1.2 Marking for America According to NEC 500

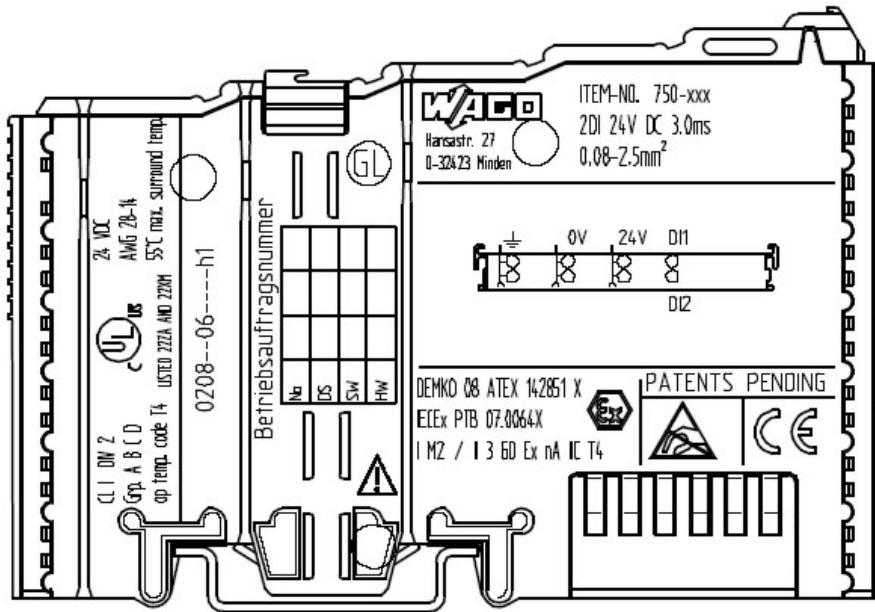


Figure 20: Side marking example for I/O modules according to NEC 500



Figure 21: Text detail – Marking example for I/O modules according to NEC 500

Table 20: Description of marking example for I/O modules according to NEC 500

Printing on Text	Description
CL 1	Explosion protection group (condition of use category)
DIV 2	Area of application (zone)
Grp. ABCD	Explosion group (gas group)
Optemp code T4	Temperature class

8.2 Installation Regulations

In the **Federal Republic of Germany**, various national regulations for the installation in explosive areas must be taken into consideration. The basis for this forms the working reliability regulation, which is the national conversion of the European guideline 99/92/E6. They are complemented by the installation regulation EN 60079-14. The following are excerpts from additional VDE regulations:

Table 21: VDE installation regulations in Germany

DIN VDE 0100	Installation in power plants with rated voltages up to 1000 V
DIN VDE 0101	Installation in power plants with rated voltages above 1 kV
DIN VDE 0800	Installation and operation in telecommunication plants including information processing equipment
DIN VDE 0185	lightning protection systems

The **USA** and **Canada** have their own regulations. The following are excerpts from these regulations:

Table 22: Installation regulations in USA and Canada

NFPA 70	National Electrical Code Art. 500 Hazardous Locations
ANSI/ISA-RP 12.6-1987	Recommended Practice
C22.1	Canadian Electrical Code

NOTICE

Notice the following points

When using the **WAGO-I/O SYSTEM 750** (electrical operation) with Ex approval, the following points are mandatory:

8.2.1 Special Conditions for Safe Operation of the ATEX and IEC Ex (acc. DEMKO 08 ATEX 142851X and IECEx PTB 07.0064)

The fieldbus-independent I/O modules of the WAGO-I/O-SYSTEM 750-.../...-... must be installed in an environment with degree of pollution 2 or better. In the final application, the I/O modules must be mounted in an enclosure with IP 54 degree of protection at a minimum with the following exceptions:

- I/O modules 750-440, 750-609 and 750-611 must be installed in an IP 64 minimum enclosure.
- I/O module 750-540 must be installed in an IP 64 minimum enclosure for 230 V AC applications.
- I/O module 750-440 may be used up to max. 120 V AC.

When used in the presence of combustible dust, all devices and the enclosure shall be fully tested and assessed in compliance with the requirements of IEC 61241-0:2004 and IEC 61241-1:2004.

When used in mining applications the equipment shall be installed in a suitable enclosure according to EN 60079-0:2006 and EN 60079-1:2007.

I/O modules fieldbus plugs or fuses may only be installed, added, removed or replaced when the system and field supply is switched off or the area exhibits no explosive atmosphere.

DIP switches, coding switches and potentiometers that are connected to the I/O module may only be operated if an explosive atmosphere can be ruled out.

I/O module 750-642 may only be used in conjunction with antenna 758-910 with a max. cable length of 2.5 m.

To exceed the rated voltage no more than 40%, the supply connections must have transient protection.

The permissible ambient temperature range is 0 °C to +55 °C.

8.2.2 Special conditions for safe use (ATEX Certificate TÜV 07 ATEX 554086 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the field bus independent I/O modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15, EN 61241-0 and EN 61241-1. For use as group I, electrical apparatus M2, the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.
2. If the interface circuits are operated without the field bus coupler station type 750-3../...-... (DEMKO 08 ATEX 142851 X), measures must be taken outside of the device so that the rating voltage is not being exceeded of more than 40% because of transient disturbances.
3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “CF-Card”, “USB”, “Fieldbus connection“, “Configuration and programming interface“, “antenna socket“, “D-Sub“ and the “Ethernet interface“. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.
6. For the type 750-601 the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The ambient temperature range is: $0^{\circ}\text{C} \leq T_a \leq +55^{\circ}\text{C}$ (for extended details please note certificate).

8.2.3 Special conditions for safe use (IEC-Ex Certificate TUN 09.0001 X)

1. For use as Dc- or Gc-apparatus (in zone 2 or 22) the fieldbus independent I/O modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15, IEC 61241-0 and IEC 61241-1. For use as group I, electrical apparatus M2, the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40% because of transient disturbances.
3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces "CF-Card", "USB", "Fieldbus connection", "Configuration and programming interface", "antenna socket", "D-Sub" and the "Ethernet interface". These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.
6. For the type 750-601 the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The ambient temperature range is: $0^{\circ}\text{C} \leq T_a \leq +55^{\circ}\text{C}$ (For extensions please see the certificate).

8.2.4 Special conditions for safe use (ATEX Certificate DEKRA 11ATEX0203 X)

1. The components shall be installed in a suitable enclosure providing a degree of protection of at least IP54 according to EN 60529, taking into account the environmental conditions under which the equipment will be used.
2. When the temperature under rated conditions exceeds 70 °C at the cable or conduit entry point, or 80 °C at the branching point of the conductors, the temperature specification of the selected cable shall be in compliance with the actual measured temperature values.
3. Provisions shall be made to prevent the rated voltages from being exceeded by transient disturbances of more than 40 %.
4. Components may only be removed or inserted when the system supply and the field supply are switched off, or when the location is known to be non-hazardous.
5. Fieldbus connectors may only be disconnected or connected when the system supply is switched off, or when the location is known to be non-hazardous.
6. The fuses, present in de the system modules 750-6xx may only be replaced when the supply is switched off, or when the location is known to be non-hazardous.
7. Address selectors and ID switches may only be adjusted when the system supply is switched off, or when the location is known to be non-hazardous.
8. The ambient temperature range is: $0^{\circ}\text{C} \leq T_a \leq +55^{\circ}\text{C}$.

8.2.5 ANSI/ISA 12.12.01

This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only.

This equipment is to be fitted within tool-secured enclosures only.

WARNING

Explosion hazard!

Explosion hazard - substitution of components may impair suitability for Class I, Div. 2.

WARNING

Disconnect device when power is off and only in a non-hazardous area!

Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous near each operator accessible connector and fuse holder." When a fuse is provided, the following information shall be provided: "A switch suitable for the location where the equipment is installed shall be provided to remove the power from the fuse."

For devices with ETHERNET connectors:

"Only for use in LAN, not for connection to telecommunication circuits".

WARNING

Use only with antenna module 758-910!

Use Module 750-642 only with antenna module 758-910.

For Couplers/Controllers and Economy bus modules only: "The configuration Interface Service connector is for temporary connection only. Do not connect or disconnect unless the area is known to be nonhazardous. Connection or disconnection in an explosive atmosphere could result in an explosion.

WARNING

Devices containing fuses must not be fitted into circuits subject to over loads!

Devices containing fuses must not be fitted into circuits subject to over loads, e.g. motor circuits!

 **WARNING**

For devices equipped with SD card slots: Insert or remove the SD cards unless the area known to be free of ignitable concentrations of flammable gases or vapors!

Do not connect or disconnect SD-Card while circuit is live unless the area is known to be free of ignitable concentrations of flammable gases or vapors.

 **Information****Additional Information**

Proof of certification is available on request.

Also take note of the information given on the operating and assembly instructions.

The manual, containing these special conditions for safe use, must be readily available to the user.

List of Figures

Figure 1: Internal Data Bus Structure with Bus Extension (example) 13

Figure 2: Isolation coupler module (example) 16

Figure 3: View 17

Figure 4: Data Contacts 18

Figure 5: Power Jumper Contacts 19

Figure 6: CAGE CLAMP® Connectors 20

Figure 7: Connectors 21

Figure 8: Display Elements – Power supply 22

Figure 9: Display Elements – Bus connection 22

Figure 10: Operating Elements 23

Figure 11: Schematic diagram 23

Figure 12: Insert I/O Module (Example) 28

Figure 13: Snap the I/O Module into Place (Example) 28

Figure 14: Removing the I/O Module (Example) 29

Figure 15: Connecting a Conductor to a CAGE CLAMP® 31

Figure 16: Side marking example for ATEX and IEC Ex approved I/O modules
according to CENELEC and IEC 34

Figure 17: Printing Text detail – Marking example for ATEX and IEC Ex
approved I/O modules according to CENELEC and IEC 34

Figure 18: Side marking example for Ex i and IEC Ex i approved I/O modules
according to CENELEC and IEC 35

Figure 19: Text detail – Marking example for Ex i and IEC Ex i approved I/O
modules according to CENELEC and IEC 35

Figure 20: Side marking example for I/O modules according to NEC 500 37

Figure 21: Text detail – Marking example for I/O modules according to NEC 500
..... 37

List of Tables

Table 1: Number Notation.....	8
Table 2: Font Conventions	8
Table 3: Legend for the “View” figure	17
Table 4: Legend for the “Power Jumper Contacts” Figure	19
Table 5: Legend for the „CAGE CLAMP®-Connectors“ Figure	20
Table 6: Legend to the „Connectors“ Figure.....	21
Table 7: Legend for „Display Elements – Power supply“	22
Table 8: Legend for „Display Elements – Bus connection“	22
Table 9: Legend for the „Operating Elements “ figure	23
Table 10: Technical Data – Device	24
Table 11: Technical Data – Supply	24
Table 12: Technical Data – Connection types: interface	24
Table 13: Technical Data – Connection types: field wiring.....	24
Table 14: Technical Data – Connection types: power jumper contact.....	24
Table 15: Technical Data – Connection types: internal data bus	25
Table 16: Technical Data – Communication.....	25
Table 17: Technical Data – Climatic Environmental Conditions	25
Table 18: Description of marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC.....	34
Table 19: Description of marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC.....	36
Table 20: Description of marking example for I/O modules according to NEC 500.....	37
Table 21: VDE installation regulations in Germany	38
Table 22: Installation regulations in USA and Canada	38

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: +49/5 71/8 87 – 0
Fax: +49/5 71/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>



This Page Left Intentionally Blank

WAGO END MODULE

**Operations & Maintenance Manual
December 2015**

This Page Left Intentionally Blank



WAGO-I/O-SYSTEM 750
End Module
750-600(/xxx-xxx)
Internal Bus Termination

Version 1.2.0



© 2015 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

1	Notes about this Documentation.....	5
1.1	Validity of this Documentation.....	5
1.2	Copyright.....	5
1.3	Symbols.....	6
1.4	Number Notation.....	8
1.5	Font Conventions.....	8
2	Important Notes	9
2.1	Legal Bases	9
2.1.1	Subject to Changes	9
2.1.2	Personnel Qualifications.....	9
2.1.3	Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions	9
2.1.4	Technical Condition of Specified Devices.....	10
2.2	Safety Advice (Precautions).....	11
3	Device Description	13
3.1	View	13
3.2	Connectors.....	14
3.2.1	Data Contacts/Internal Bus.....	14
3.2.2	Power Jumper Contacts/Field Supply	14
3.2.3	CAGE CLAMP® Connectors	14
3.3	Display Elements	15
3.4	Operating Elements.....	15
3.5	Technical Data	16
3.5.1	Device.....	16
3.5.2	Connection Type	16
3.5.3	Climatic Environmental Conditions.....	16
3.6	Approvals	17
3.7	Standards and Guidelines.....	20
4	Mounting.....	21
4.1	Mounting Sequence.....	21
4.2	Inserting and Removing Devices	22
4.2.1	Inserting the I/O Module	22
4.2.2	Removing the I/O Module.....	23
5	Use in Hazardous Environments	24
5.1	Marking Configuration Examples.....	25
5.1.1	Marking for Europe According to ATEX and IEC-Ex	25
5.1.2	Marking for America According to NEC 500.....	30
5.2	Installation Regulations.....	31
5.2.1	Special Conditions for Safe Use (ATEX Certificate TÜV 07 ATEX 554086 X).....	32
5.2.2	Special Conditions for Safe Use (ATEX Certificate TÜV 12 ATEX 106032 X).....	33
5.2.3	Special Conditions for Safe Use (IEC-Ex Certificate TUN 09.0001 X)34	
5.2.4	Special Conditions for Safe Use (IEC-Ex Certificate IECEX TUN 12.0039 X).....	35

5.2.5 Special Conditions for Safe Use According to ANSI/ISA 12.12.01 .. 36

List of Figures 37

List of Tables..... 38

1 Notes about this Documentation

Note



Always retain this documentation!

This documentation is part of the product. Therefore, retain the documentation during the entire service life of the product. Pass on the documentation to any subsequent user. In addition, ensure that any supplement to this documentation is included, if necessary.

1.1 Validity of this Documentation

This documentation is only applicable to the I/O module 750-600 (End Module) and the variants listed in the table below.

Table 1: Variants

Item Number/Variant	Designation
750-600	End Module
750-600/025-000	End Module/T

Note



Documentation Validity for Variants

Unless otherwise indicated, the information given in this documentation applies to listed variants.

The I/O module 750-600 shall only be installed and operated according to the instructions in this manual and in the manual for the used fieldbus coupler/controller.

NOTICE

Consider power layout of the WAGO-I/O-SYSTEM 750!

In addition to these operating instructions, you will also need the manual for the used fieldbus coupler/controller, which can be downloaded at www.wago.com. There, you can obtain important information including information on electrical isolation, system power and supply specifications.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.3 Symbols

DANGER

Personal Injury!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

DANGER



Personal Injury Caused by Electric Current!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

Personal Injury!

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION

Personal Injury!

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE



Damage to Property Caused by Electrostatic Discharge (ESD)!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note



Important Note!

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.



Information

Additional Information:

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.4 Number Notation

Table 2: Number Notation

Number Code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.5 Font Conventions

Table 3: Font Conventions

Font Type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Program Files\WAGO Software</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualifications

All sequences implemented on WAGO-I/O-SYSTEM 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

All changes to the coupler or controller should always be carried out by qualified personnel with sufficient skills in PLC programming.

2.1.3 Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions

Fieldbus couplers, fieldbus controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to actuators or higher-level control systems. Using programmable controllers, the signals can also be (pre-) processed.

The devices have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the devices in wet and dusty environments is prohibited.

Operating the WAGO-I/O-SYSTEM 750 devices in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section “Device Description” > “Standards and Guidelines” in the manual for the used fieldbus coupler/controller.

Appropriate housing (per 94/9/EG) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

2.1.4 Technical Condition of Specified Devices

The devices to be supplied ex works are equipped with hardware and software configurations, which meet the individual application requirements. WAGO Kontakttechnik GmbH & Co. KG will be exempted from any liability in case of changes in hardware or software as well as to non-compliant usage of devices.

Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

2.2 Safety Advice (Precautions)

For installing and operating purposes of the relevant device to your system the following safety precautions shall be observed:



DANGER

Do not work on devices while energized!

All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.

DANGER

Install the device only in appropriate housings, cabinets or in electrical operation rooms!

The WAGO-I/O-SYSTEM 750 and its components are an open system. As such, install the system and its components exclusively in appropriate housings, cabinets or in electrical operation rooms. Allow access to such equipment and fixtures to authorized, qualified staff only by means of specific keys or tools.

NOTICE

Replace defective or damaged devices!

Replace defective or damaged device/module (e.g., in the event of deformed contacts), since the long-term functionality of device/module involved can no longer be ensured.

NOTICE

Protect the components against materials having seeping and insulating properties!

The components are not resistant to materials having seeping and insulating properties such as: aerosols, silicones and triglycerides (found in some hand creams). If you cannot exclude that such materials will appear in the component environment, then install the components in an enclosure being resistant to the above-mentioned materials. Clean tools and materials are imperative for handling devices/modules.

NOTICE

Clean only with permitted materials!

Clean soiled contacts using oil-free compressed air or with ethyl alcohol and leather cloths.

NOTICE**Do not use any contact spray!**

Do not use any contact spray. The spray may impair contact area functionality in connection with contamination.

NOTICE**Do not reverse the polarity of connection lines!**

Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

NOTICE**Avoid electrostatic discharge!**

The devices are equipped with electronic components that may be destroyed by electrostatic discharge when touched. Please observe the safety precautions against electrostatic discharge per DIN EN 61340-5-1/-3. When handling the devices, please ensure that environmental factors (personnel, work space and packaging) are properly grounded.

3 Device Description

The module 750-600 (End Module) is used to terminate the internal bus of a fieldbus node. This module completes the internal data circuit and ensures correct data flow.

Note



Don't forget the bus end module!

Always plug a bus end module 750-600 onto the end of the fieldbus node! You must always use a bus end module at all fieldbus nodes with WAGO-I/O-SYSTEM 750 fieldbus couplers/controllers to guarantee proper data transfer.

3.1 View

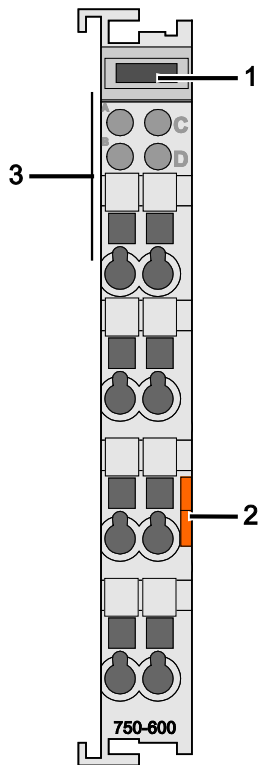


Figure 1: View

Table 4: Legend for Figure “View”

Pos.	Description	Details see section
1	Marking possibility with Mini-WSB	---
2	Release tab	“Mounting” > “Inserting and Removing Devices”
3	Data contacts	“Device Description” > “Connectors”

3.2 Connectors

3.2.1 Data Contacts/Internal Bus

Communication between the fieldbus coupler/controller and the I/O modules as well as the system supply of the I/O modules is carried out via the internal bus. It is comprised of 6 data contacts, which are available as self-cleaning gold spring contacts.

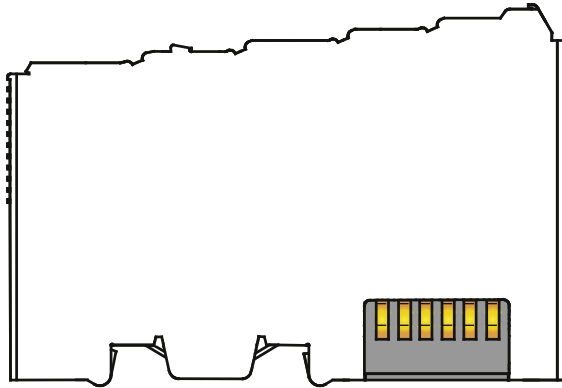


Figure 2: Data Contacts

NOTICE

Do not place the I/O modules on the gold spring contacts!

Do not place the I/O modules on the gold spring contacts in order to avoid soiling or scratching!

NOTICE



Ensure that the environment is well grounded!

The devices are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the devices, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, e.g. data contacts.

3.2.2 Power Jumper Contacts/Field Supply

The I/O module 750-600 has no power jumper contacts.

3.2.3 CAGE CLAMP® Connectors

The I/O module 750-600 has no CAGE CLAMP® connectors.

3.3 Display Elements

The I/O module 750-600 has no display elements.

3.4 Operating Elements

The I/O module 750-600 has no operating elements.

3.5 Technical Data

3.5.1 Device

Table 5: Technical Data – Device

Width	12 mm
Height (from upper edge of 35 DIN rail)	64 mm
Depth	100 mm
Weight	Approx. 35 g

3.5.2 Connection Type

Table 6: Technical Data – Data Contacts

Data contacts	Slide contact, hard gold plated, self-cleaning
---------------	--

3.5.3 Climatic Environmental Conditions

Table 7: Technical Data – Climatic Environmental Conditions

Operating temperature range	0 °C ... 55 °C
Operating temperature range for components with extended temperature range (750-xxx/025-xxx)	-20 °C ... +60 °C
Storage temperature range	-25 °C ... +85 °C
Storage temperature range for components with extended temperature range (750-xxx/025-xxx)	-40 °C ... +85 °C
Relative humidity	Max. 5 % ... 95 % without condensation
Resistance to harmful substances	Acc. to IEC 60068-2-42 and IEC 60068-2-43
Maximum pollutant concentration at relative humidity < 75 %	SO ₂ ≤ 25 ppm H ₂ S ≤ 10 ppm
Special conditions	Ensure that additional measures for components are taken, which are used in an environment involving: <ul style="list-style-type: none"> – dust, caustic vapors or gases – ionizing radiation

3.6 Approvals

Information




More information about approvals.

Detailed references to the approvals are listed in the document “Overview Approvals **WAGO-I/O-SYSTEM 750**”, which you can find via the internet under: www.wago.com > SERVICES > DOWNLOADS > Additional documentation and information on automation products > WAGO-I/O-SYSTEM 750 > System Description.


The following approvals have been granted to the basic version and all variants of 750-600 I/O modules:

 Conformity Marking

 cUL_{US} UL508

 Korea Certification MSIP-REM-W43-END750

The following Ex approvals have been granted to the basic version and all variants of 750-600 I/O modules:

 TÜV 07 ATEX 554086 X
I M2 Ex d I Mb
II 3 G Ex nA IIC T4 Gc
II 3 D Ex tc IIIC T135°C Dc

Permissible ambient temperature range:

- Standard: $0\text{ °C} \leq T_a \leq +60\text{ °C}$
- Variants with extended temperature range (750-xxx/025-xxx): $-20\text{ °C} \leq T_a \leq +60\text{ °C}$

IECEX TUN 09.0001 X

Ex d I Mb
Ex nA IIC T4 Gc
Ex tc IIIC T135°C Dc

Permissible ambient temperature range:











- Standard: $0\text{ °C} \leq T_a \leq +60\text{ °C}$
- Variants with extended temperature range (750-xxx/025-xxx): $-20\text{ °C} \leq T_a \leq +60\text{ °C}$

The following Ex approvals have been granted to the basic version of 750-600 I/O modules:

 cULus ANSI/ISA 12.12.01
Class I, Div2 ABCD T4

The following ship approvals have been granted to the basic version and all variants of 750-600 I/O modules listed in the table:

Table 8: Ship Approvals

										
750-600	X	X	X	X	X	X	X	X	X	X
750-600 /025-000		X			X					



ABS (American Bureau of Shipping)



BV (Bureau Veritas)



DNV (Det Norske Veritas) Class B



GL (Germanischer Lloyd) Cat. A, B, C, D (EMC 1)



KR (Korean Register of Shipping)



LR (Lloyd's Register) Env. 1, 2, 3, 4



NKK (Nippon Kaiji Kyokai)



PRS (Polski Rejestr Statków)



RINA (Registro Italiano Navale)

3.7 Standards and Guidelines

All variations of 750-600 I/O modules meet the following requirements on emission and immunity of interference:

EMC CE-Emission of interference acc. to EN 61000-6-3

EMC CE-Immunity to interference acc. to EN 61000-6-2

EMC marine applications-Emission of interference acc. to Germanischer Lloyd

EMC marine applications-Immunity to interference acc. to Germanischer Lloyd

4 Mounting

4.1 Mounting Sequence

Fieldbus couplers/controllers and I/O modules of the WAGO-I/O-SYSTEM 750/753 are snapped directly on a carrier rail in accordance with the European standard EN 50022 (DIN 35).

The reliable positioning and connection is made using a tongue and groove system. Due to the automatic locking, the individual devices are securely seated on the rail after installation.

Starting with the fieldbus coupler/controller, the I/O modules are mounted adjacent to each other according to the project design. Errors in the design of the node in terms of the potential groups (connection via the power contacts) are recognized, as the I/O modules with power contacts (blade contacts) cannot be linked to I/O modules with fewer power contacts.

CAUTION

Risk of injury due to sharp-edged blade contacts!

The blade contacts are sharp-edged. Handle the I/O module carefully to prevent injury.

NOTICE

Insert I/O modules only from the proper direction!

All I/O modules feature grooves for power jumper contacts on the right side. For some I/O modules, the grooves are closed on the top. Therefore, I/O modules featuring a power jumper contact on the left side cannot be snapped from the top. This mechanical coding helps to avoid configuration errors, which may destroy the I/O modules. Therefore, insert I/O modules only from the right and from the top.

Note



Don't forget the bus end module!

Always plug a bus end module 750-600 onto the end of the fieldbus node! You must always use a bus end module at all fieldbus nodes with WAGO-I/O-SYSTEM 750 fieldbus couplers/controllers to guarantee proper data transfer.

4.2 Inserting and Removing Devices

NOTICE

Perform work on devices only if they are de-energized!

Working on energized devices can damage them. Therefore, turn off the power supply before working on the devices.

4.2.1 Inserting the I/O Module

1. Position the I/O module so that the tongue and groove joints to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are engaged.

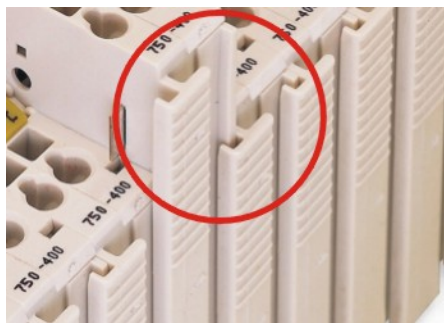


Figure 3: Insert I/O Module (Example)

2. Press the I/O module into the assembly until the I/O module snaps into the carrier rail.

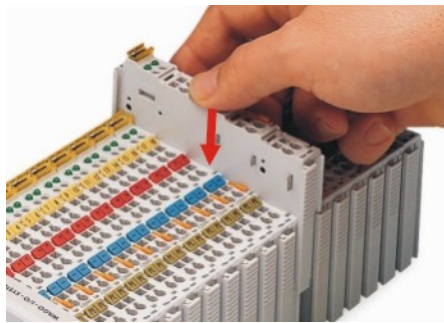


Figure 4: Snap the I/O Module into Place (Example)

With the I/O module snapped in place, the electrical connections for the data contacts and power jumper contacts (if any) to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are established.

4.2.2 Removing the I/O Module

1. Remove the I/O module from the assembly by pulling the release tab.

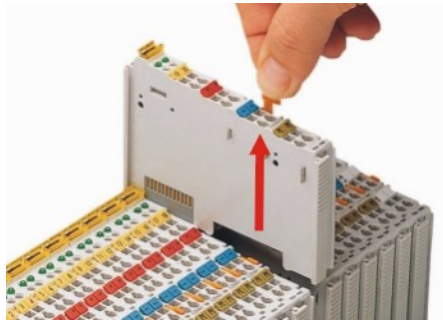


Figure 5: Removing the I/O Module (Example)

Electrical connections for data or power jumper contacts are disconnected when removing the I/O module.

5 Use in Hazardous Environments

The **WAGO-I/O-SYSTEM 750** (electrical equipment) is designed for use in Zone 2 hazardous areas.

The following sections include both the general identification of components (devices) and the installation regulations to be observed. The individual subsections of the “Installation Regulations” section must be taken into account if the I/O module has the required approval or is subject to the range of application of the ATEX directive.

5.1 Marking Configuration Examples

5.1.1 Marking for Europe According to ATEX and IEC-Ex

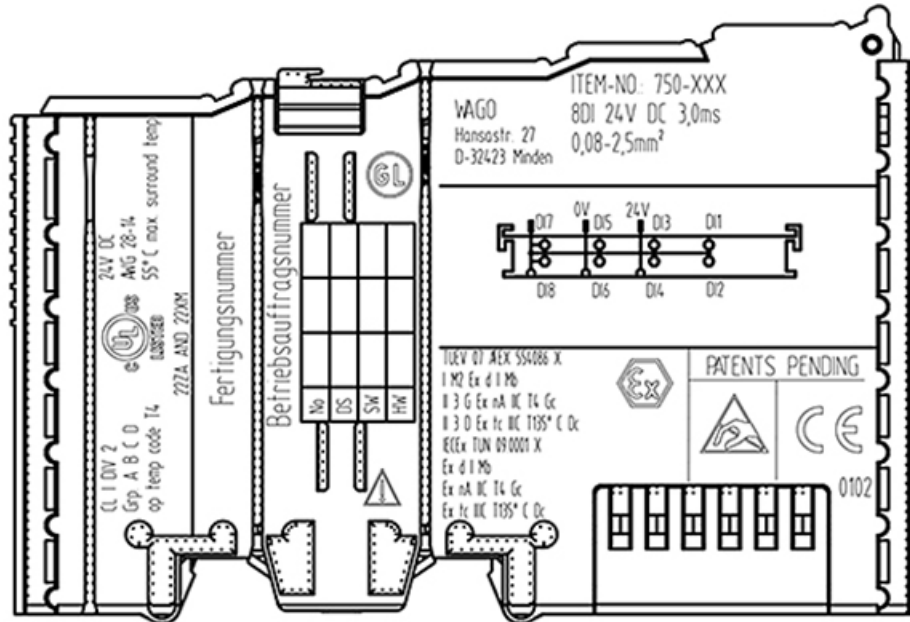


Figure 6: Side Marking Example for Approved I/O Modules According to ATEX and IECEx

TUEV 07 AEX 554086 X
I M2 Ex d I Mb
II 3 G Ex nA IIC T4 Gc
II 3 D Ex tc IIC T135° C Dc
IECEx TUN 09.0001 X
Ex d I Mb
Ex nA IIC T4 Gc
Ex tc IIC T135° C Dc




Figure 7: Text Detail – Marking Example for Approved I/O Modules According to ATEX and IECEx.

Table 9: Description of Marking Example for Approved I/O Modules According to ATEX and IECEx

Printing on Text	Description
TÜV 07 ATEX 554086 X IECEx TUN 09.0001 X	Approving authority and certificate numbers
Dust	
II	Equipment group: All except mining
3D	Category 3 (Zone 22)
Ex	Explosion protection mark
tc Dc	Type of protection and equipment protection level (EPL): protection by enclosure
IIIC	Explosion group of dust
T 135°C	Max. surface temperature of the enclosure (without a dust layer)
Mining	
I	Equipment group: Mining
M2	Category: High level of protection
Ex	Explosion protection mark
d Mb	Type of protection and equipment protection level (EPL): Flameproof enclosure
I	Explosion group for electrical equipment for mines susceptible to firedamp
Gases	
II	Equipment group: All except mining
3G	Category 3 (Zone 2)
Ex	Explosion protection mark
nA Gc	Type of protection and equipment protection level (EPL): Non-sparking equipment
nC Gc	Type of protection and equipment protection level (EPL): Sparking apparatus with protected contacts. A device which is so constructed that the external atmosphere cannot gain access to the interior
IIIC	Explosion group of gas and vapours
T4	Temperature class: Max. surface temperature 135°C

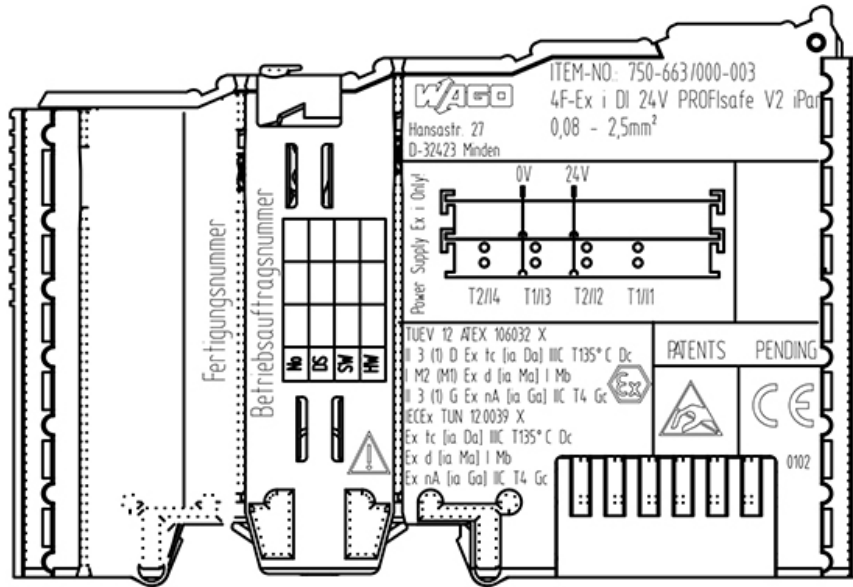


Figure 8: Side Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx.


TUEV 12 ATEX 106032 X
 II 3 (1) D Ex tc [ia Da] IIC T135° C Dc
 I M2 (M1) Ex d [ia Ma] I Mb
 II 3 (1) G Ex nA [ia Ga] IIC T4 Gc 
 IECEx TUN 12.0039 X
 Ex tc [ia Da] IIC T135° C Dc
 Ex d [ia Ma] I Mb
 Ex nA [ia Ga] IIC T4 Gc

Figure 9: Text Detail – Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx.

Table 10: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx

Inscription Text	Description
TÜV 07 ATEX 554086 X IECEx TUN 09.0001X	Approving authority and certificate numbers
TÜV 12 ATEX 106032 X IECEx TUN 12.0039 X	
Dust	
II	Equipment group: All except mining
3(1)D	Category 3 (Zone 22) equipment containing a safety device for a category 1 (Zone 20) equipment
3(2)D	Category 3 (Zone 22) equipment containing a safety device for a category 2 (Zone 21) equipment
Ex	Explosion protection mark
tc Dc	Type of protection and equipment protection level (EPL): protection by enclosure
[ia Da]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 20
[ib Db]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 21
IIIC	Explosion group of dust
T 135°C	Max. surface temperature of the enclosure (without a dust layer)
Mining	
I	Equipment Group: Mining
M2 (M1)	Category: High level of protection with electrical circuits which present a very high level of protection
Ex d Mb	Explosion protection mark with Type of protection and equipment protection level (EPL): Flameproof enclosure
[ia Ma]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety electrical circuits
I	Explosion group for electrical equipment for mines susceptible to firedamp

Table 10: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx

Gases	
II	Equipment group: All except mining
3(1)G	Category 3 (Zone 2) equipment containing a safety device for a category 1 (Zone 0) equipment
3(2)G	Category 3 (Zone 2) equipment containing a safety device for a category 2 (Zone 1) equipment
Ex	Explosion protection mark
nA Gc	Type of protection and equipment protection level (EPL): Non-sparking equipment
[ia Ga]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 0
[ia Gb]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 1
IIC	Explosion group of gas and vapours
T4	Temperature class: Max. surface temperature 135°C

5.1.2 Marking for America According to NEC 500

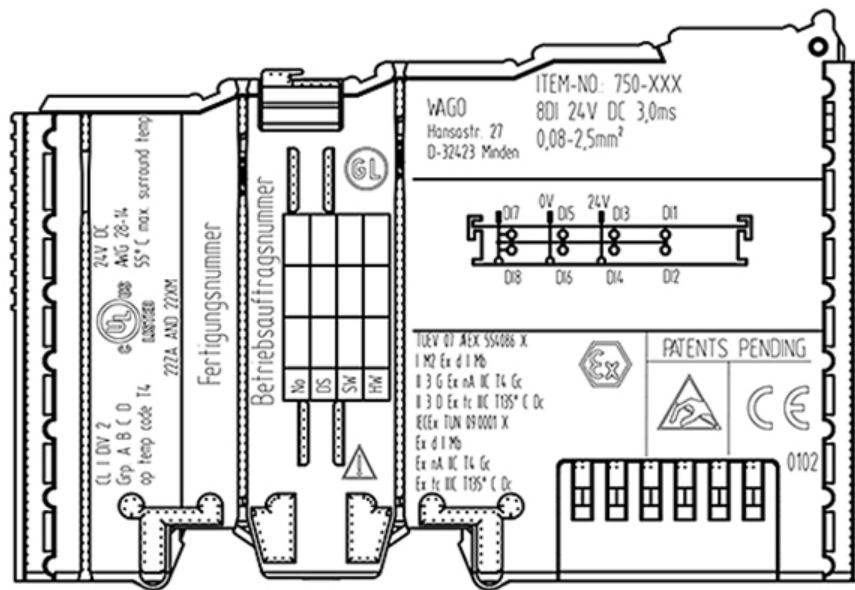


Figure 10: Side Marking Example for I/O Modules According to NEC 500

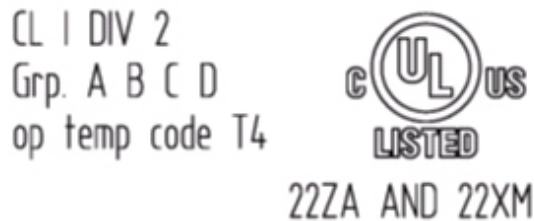


Figure 11: Text Detail – Marking Example for Approved I/O Modules According to NEC 500

Table 11: Description of Marking Example for Approved I/O Modules According to NEC 500

Printing on Text	Description
CL I	Explosion protection group (condition of use category)
DIV 2	Area of application
Grp. ABCD	Explosion group (gas group)
Op temp code T4	Temperature class

5.2 Installation Regulations

For the installation and operation of electrical equipment in hazardous areas, the valid national and international rules and regulations which are applicable at the installation location must be carefully followed.

5.2.1 Special Conditions for Safe Use (ATEX Certificate TÜV 07 ATEX 554086 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15 and EN 60079-31. For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. Dip-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “Memory-Card”, “USB”, “Fieldbus connection”, “Configuration and programming interface”, “antenna socket”, “D-Sub”, “DVI-port” and the “Ethernet interface”. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.
6. For replaceable fuses the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The following warnings shall be placed nearby the unit:
WARNING – DO NOT REMOVE OR REPLACE FUSE WHEN ENERGIZED
WARNING – DO NOT SEPARATE WHEN ENERGIZED
WARNING – SEPARATE ONLY IN A NON-HAZARDOUS AREA

5.2.2 Special Conditions for Safe Use (ATEX Certificate TÜV 12 ATEX 106032 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** Ex i shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15 and EN 60079-31. For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
4. For the type the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.

5.2.3 Special Conditions for Safe Use (IEC-Ex Certificate TUN 09.0001 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15 and IEC 60079-31. For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64.
The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “Memory-Card”, “USB”, “Fieldbus connection”, “Configuration and programming interface”, “antenna socket”, “D-Sub”, “DVI-port” and the “Ethernet interface”. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.
6. For replaceable fuses the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The following warnings shall be placed nearby the unit:
WARNING – DO NOT REMOVE OR REPLACE FUSE WHEN ENERGIZED
WARNING – DO NOT SEPARATE WHEN ENERGIZED
WARNING – SEPARATE ONLY IN A NON-HAZARDOUS AREA

5.2.4 Special Conditions for Safe Use (IEC-Ex Certificate IECEx TUN 12.0039 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus independent I/O Modules WAGO-I/O-SYSTEM 750-*** Ex i shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15, IEC 60079-31.
For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64.
The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
4. For the type the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.

5.2.5 Special Conditions for Safe Use According to ANSI/ISA 12.12.01

- A. “This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only.”
- B. “This equipment is to be fitted within tool-secured enclosures only.”
- C. “WARNING Explosion hazard - substitution of components may impair suitability for Class I, Div. 2.”
- D. “WARNING – Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous” has to be placed near each operator accessible connector and fuse holder.
- E. When a fuse is provided, the following information shall be provided: “A switch suitable for the location where the equipment is installed shall be provided to remove the power from the fuse.”
- F. For devices with EtherCAT/Ethernet connectors “Only for use in LAN, not for connection to telecommunication circuits.”
- G. “WARNING - Use Module 750-642 only with antenna module 758-910.”
- H. For Couplers/Controllers and Economy bus modules only: The instructions shall contain the following: “The configuration interface Service connector is for temporary connection only. Do not connect or disconnect unless the area is known to be non-hazardous. Connection or disconnection in an explosive atmosphere could result in an explosion.”
- I. Modules containing fuses only: “WARNING - Devices containing fuses must not be fitted into circuits subject to over loads, e.g. motor circuits.”
- J. Modules containing SD card reader sockets only: “WARNING - Do not connect or disconnect SD-Card while circuit is live unless the area is known to be free of ignitable concentrations of flammable gases or vapors.”

Information



Additional Information

Proof of certification is available on request.

Also take note of the information given on the operating and assembly instructions.

The manual, containing these special conditions for safe use, must be readily available to the user.

List of Figures

Figure 1: View 13
Figure 2: Data Contacts 14
Figure 3: Insert I/O Module (Example)..... 22
Figure 4: Snap the I/O Module into Place (Example)..... 22
Figure 5: Removing the I/O Module (Example) 23
Figure 6: Side Marking Example for Approved I/O Modules According to ATEX
and IECEX 25
Figure 7: Text Detail – Marking Example for Approved I/O Modules According
to ATEX and IECEX. 25
Figure 8: Side Marking Example for Approved Ex i I/O Modules According to
ATEX and IECEX. 27
Figure 9: Text Detail – Marking Example for Approved Ex i I/O Modules
According to ATEX and IECEX. 27
Figure 10: Side Marking Example for I/O Modules According to NEC 500 30
Figure 11: Text Detail – Marking Example for Approved I/O Modules According
to NEC 500 30

List of Tables

Table 1: Variants	5
Table 2: Number Notation.....	8
Table 3: Font Conventions	8
Table 4: Legend for Figure “View”	13
Table 5: Technical Data – Device	16
Table 6: Technical Data – Data Contacts	16
Table 7: Technical Data – Climatic Environmental Conditions	16
Table 8: Ship Approvals.....	19
Table 9: Description of Marking Example for Approved I/O Modules According to ATEX and IECEx	26
Table 10: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx	28
Table 11: Description of Marking Example for Approved I/O Modules According to NEC 500	30

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: +49/5 71/8 87 – 0
Fax: +49/5 71/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>



This Page Left Intentionally Blank

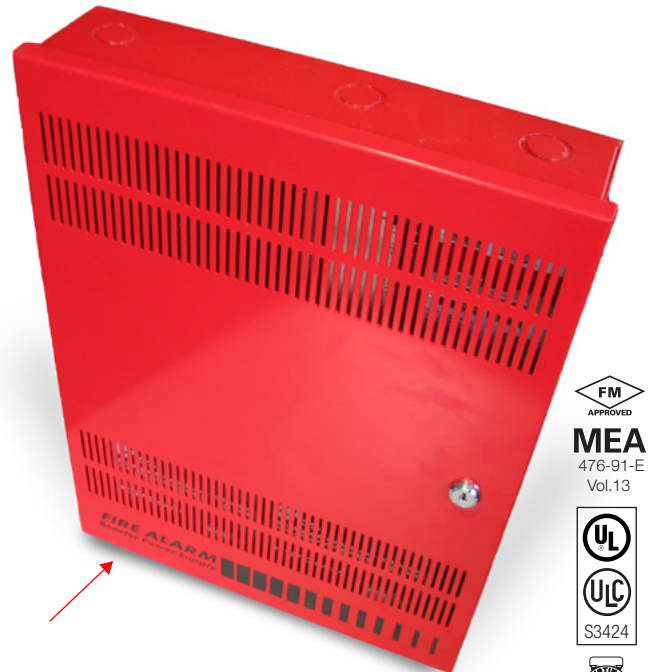
REMOTE BOOSTER POWER
SUPPLY, 10-AMP (LIOS
DETECTOR)

Operations & Maintenance Manual
December 2015



Remote Booster Power Supplies

BPS6A, BPS10A



ME A
476-91-E
Vol.13



ULC
S3424



7300-1657:
0229

Overview

The Booster Power Supply (BPS) is a UL 864, 9th Edition listed power supply. It is a 24 Vdc filtered-regulated, and supervised unit that can easily be configured to provide additional notification appliance circuits (NACs) or auxiliary power for Mass Notification/Emergency Communication (MNEC), as well as life safety, security, and access control applications.

The BPS contains the circuitry to monitor and charge internal or external batteries. Its steel enclosure has room for up to two 10 ampere-hour batteries. For access control-only applications, the BPS can support batteries totaling up to 65 ampere-hours in an external enclosure. The BPS has four Class B (convertible to two Class A) NACs. These can be activated in one or two groups from the BPS's unique dual input circuits.

The BPS is available in 6.5 or 10 ampere models. Each output circuit has a capacity of three amperes; total current draw cannot exceed the unit's rating.

The BPS meets current UL requirements and is listed under the following standards:

Standard (CCN)	Description
UL864 9th ed.ition (UOXX)	Fire Alarm Systems
UL636 (ANET, UEHX7)	Holdup Alarm Units and Systems
UL609 (AOTX, AOTX7)	Local Burglar Alarm Units and Systems
UL294 (ALVY, UEHX7)	Access Control Systems
UL365 (APAW, APAW7)	Police Station Connected Burglar Alarm Units and Systems
UL1076 (APOU, APOU7)	Proprietary Burglar Alarm System Units
UL1610 (AMCX)	Central Station Alarm Unit
ULC-S527 (UOXXC)	Control Units, Fire Alarm (Canada)
ULC-S303 (AOTX7)	Local Burglar Alarm Units and Systems (Canada)
C22.2 No. 205	Signaling Equipment (Canada)

Standard Features

- Allows for reliable filtered and regulated power to be installed where needed
- Cost effective system expansion
- Provides for Genesis and Enhanced Integrity notification appliance synchronization
- Supports coded output operation
- Self-restoring overcurrent protection
- Multiple signal rates
- Can be cascaded or controlled independently
- Easy field configuration
- On-board diagnostic LEDs identify wiring or internal faults
- Standard Edwards keyed lockable steel cabinet with removable door
- 110 and 230 Vac models available
- Accommodates 18 to 12 AWG wire sizes
- Optional tamper switch
- Dual battery charging rates
- Optional earthquake hardening; OSHPD seismic pre-approval for component Importance Factor 1.5

Application

The BPS provides additional power and circuits for notification appliances and other 24 Vdc loads. It is listed for indoor dry locations and can easily be installed where needed.

Fault conditions are indicated on the on-board diagnostic LEDs, opening the BPS input sense circuit and the trouble relay (if programmed). While this provides indication to the host system, the BPS can still be activated upon command. A separate AC Fail contact is available on the BPS circuit board, which can be programmed for trouble or AC Fail. There are seven on-board diagnostic LEDs: one for each NAC fault, one for battery fault, one for ground fault, and one for AC power.

The unique dual-input activation circuits of the BPS can be activated by any voltage from 6 to 45 VDC (filtered-regulated) or 11 to 33 Vdc (full-wave rectified, unfiltered). The first input circuit can be configured to activate 1-4 of the four possible outputs. The second input circuit can be configured to control circuits 3 and 4. When outputs are configured for auxiliary operation, these circuits can be configured to stay on or automatically deactivate 30 seconds after AC power is lost. This feature makes these circuits ideal for door holder applications. The BPS also has a separate 200 mA 24 Vdc output that can be used to power internal activation modules.

BPS NACs can be configured for a 3-3-3 temporal or continuous output. California temporal rate outputs are also available on certain models. This makes the BPS ideal for applications requiring signaling rates that are not available from the main system.

In addition to the internally generated signal rates, the BPS can also be configured to follow the coded signal rate of the main system NACs. This allows for the seamless expansion of existing NACs.

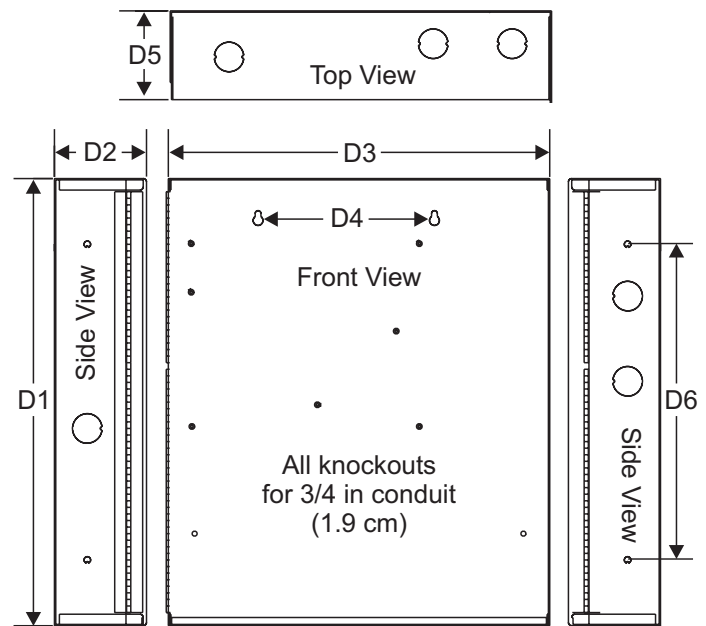
The BPS enclosure has mounting brackets for up to three Signature modules to the right of the circuit board.

Engineering Specification

Supply, where needed, Edwards BPS Series Booster Power Supplies (BPS) that are interconnected to and supervised by the main system. The BPS shall function as a stand-alone auxiliary power supply with its own fully-supervised battery compliment. The BPS battery compliment shall be sized to match the requirements of the main system. The BPS shall be capable of supervising and charging batteries having the capacity of 24 ampere-hours for Mass Notification/Emergency Communication (MNEC), life safety and security applications, and the capacity of 65 ampere-hours for access control applications.

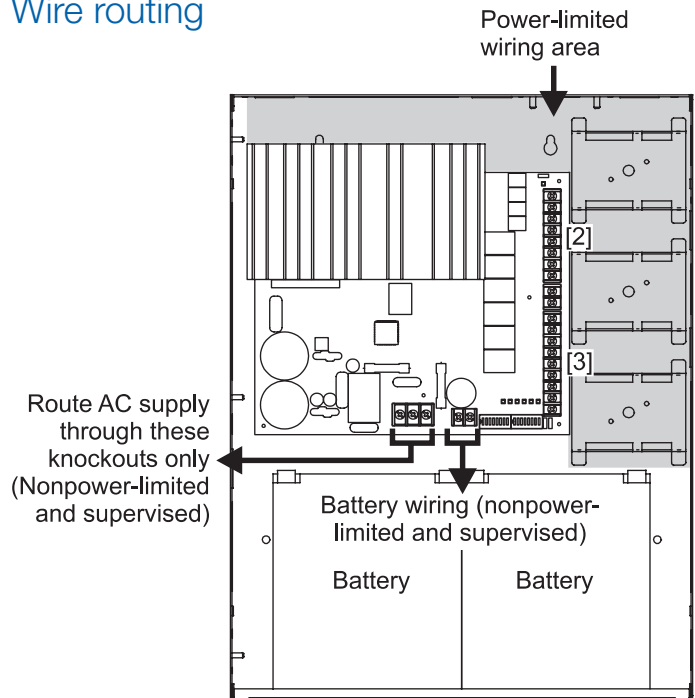
<<The BPS shall be capable of installation for a seismic component Importance Factor of 1.5.>> The BPS shall provide a minimum of four independent, fully supervised Class B circuits that can be field configurable for notification appliance circuits or auxiliary 24 Vdc power circuits. BPS NACs shall be convertible to a minimum of two Class A NACs. Each BPS output circuit shall be rated at 3 amperes at 24 Vdc. Each output circuit shall be provided with automatically restoring overcurrent protection. The BPS shall be operable from the main system NAC and/or Edwards Signature Series control modules. BPS NACs shall be configurable for continuous, 3-3-3 temporal or optionally, California rate. Fault conditions on the BPS shall not impede operation of main system NAC. The BPS shall be provided with ground fault detection circuitry and a separate AC fail relay.

Dimensions



D1	D2	D3	D4	D5	D6
17.0 in (43.2 cm)	3.5 in (8.9 cm)	13.0 in (33.0 cm)	6.5 in (16.5 cm)	3.375 in (8.6 cm)	12.0 in (30.4 cm)

Wire routing



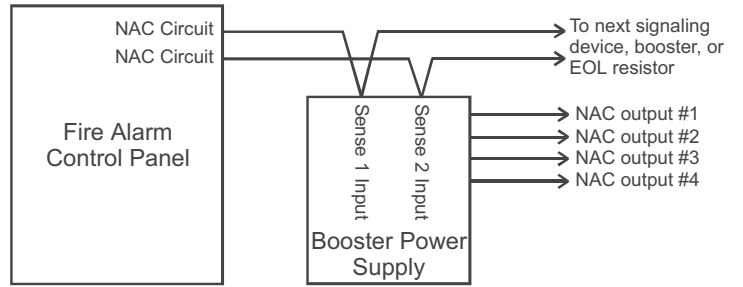
Notes

1. Maintain 1/4-inch (6 mm) spacing between power-limited and nonpower-limited wiring or use type FPL, FPLR, or FPLP cable per NEC.
2. Power-limited and supervised when not configured as auxiliary power. Non-supervised when configured as auxiliary power.
3. Source must be power-limited. Source determines supervision.
4. When using larger batteries, make sure to position the battery terminals towards the door.

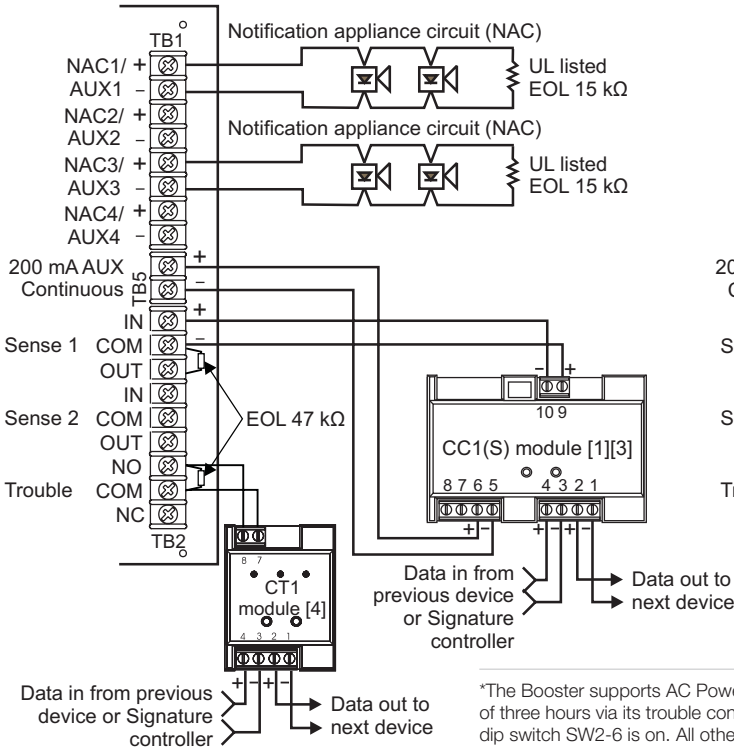
Typical Wiring

Single or cascaded booster anywhere on a notification appliance circuit

Existing NAC end-of-line resistors are not required to be installed at the booster's terminals. This allows multiple boosters to be driven from a single NAC circuit without the need for special configurations.

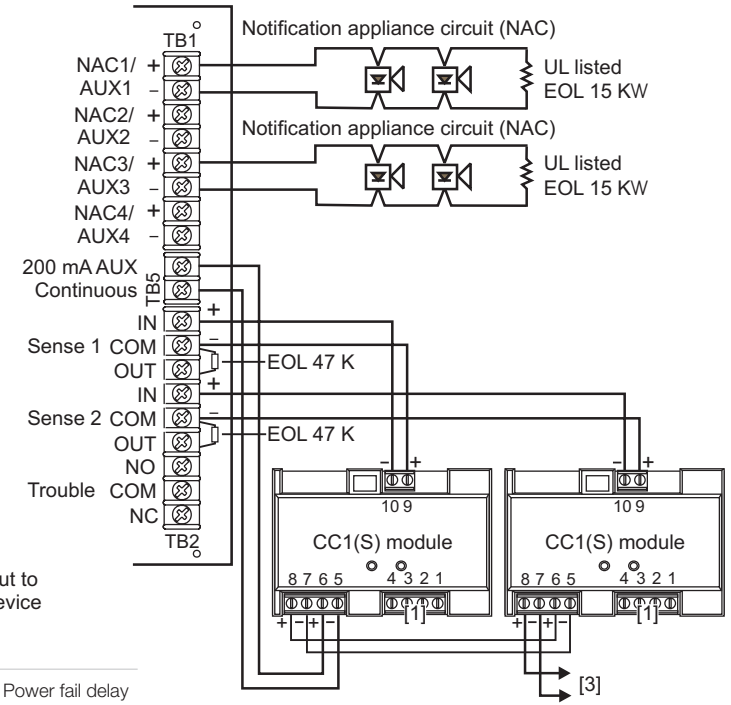


Configuring the Booster for AC Power Fail delay operation*

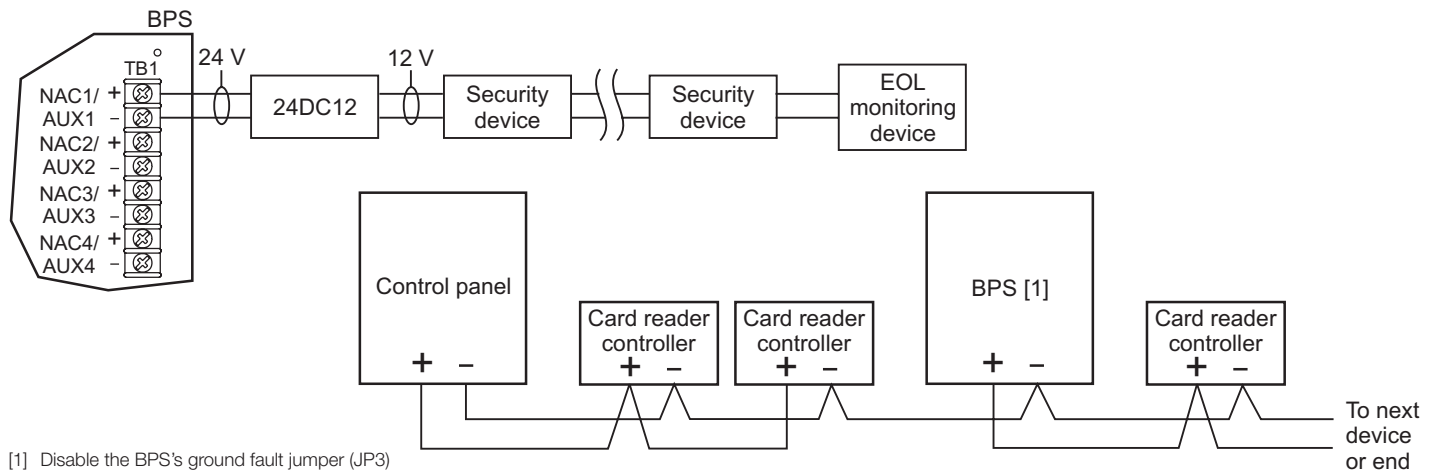


*The Booster supports AC Power fail delay of three hours via its trouble contact when dip switch SW2-6 is on. All other troubles are reported to supervising module or panel without delay via Sense inputs.

Multiple CC1(S) modules using the BPS's sense inputs



Security and access



[1] Disable the BPS's ground fault jumper (JP3)



Contact us...

Email: edwards.fire@fs.utc.com

Web: www.est-fire.com

EST is an **EDWARDS** brand.

1016 Corporate Park Drive
Mebane, NC 27302

In Canada, contact Chubb Edwards...

Email: inquiries@chubbedwards.com

Web: www.chubbedwards.com

© 2013 UTC Fire & Security Americas Corporation, Inc. All rights reserved. Specifications subject to change without notice. Edwards is part of UTC Climate, Controls & Security, a unit of United Technologies Corporation.

Specifications

Model	6.5 amp Booster	10 amp Booster
AC Line Voltage	120VAC or 220-240VAC 50/60Hz 390 watts	120VAC or 220-240VAC 50/60Hz 580 watts
Notification Appliance Circuit Ratings	3.0A max. per circuit @ 24Vdc nominal 6.5A max total all NACs	3.0A max. per circuit @ 24Vdc nominal 10A max total all NACs
Trouble Relay	2 Amps @ 30Vdc	
Auxiliary Outputs	Four configurable outputs replace NACs 1, 2, 3 or 4. as auxiliary outputs and 200 mA dedicated auxiliary. (See note 2.)	
Input Current (from an existing NAC)	3mA @ 12Vdc, 6mA @ 24Vdc	
Booster Internal Supervisory Current	70mA + 35 mA for each circuit set to AUX	
Booster Internal Alarm Current	270mA	
Signature Mounting Space	Accommodates three two-gang modules.	
Maximum Battery Size	10 Amp Hours (2 of 12V10A) in cabinet up to 24 Amp hours with external battery cabinet for fire and security applications; up to 65 Amp hours for access control applications in external battery box.	
Terminal Wire Gauge	18-12 AWG	
Relative Humidity	0 to 93% non condensing @ 32°C	
Temperature Rating	32° to 120°F (0° to 49°C)	
NAC Wiring Styles	Class A or Class B	
Output Signal Rates	Continuous, California rate, 3-3-3 temporal, or follow installed panel's NAC. (See note 1.)	
Ground Fault Detection	Enable or Disable via jumper	
Agency Listings	UL, ULC, CSFM	

1. Model BPS*CAA provides selection for California rate, in place of temporal.
2. Maximum of 8 Amps can be used for auxiliary output.

Ordering Information

Catalog Number	Description	Shipping Wt. lb (kg)
BPS6A	6.5 Amp Booster Power Supply	13 (5.9)
BPS6AC	6.5 Amp Booster Power Supply (ULC)	13 (5.9)
BPS6A/230	6.5 Amp Booster Power Supply (220V)	13 (5.9)
BPS6CAA	6.5 Amp Booster Power Supply with California rate	13 (5.9)
BPS10A	10 Amp Booster Power Supply	13 (5.9)
BPS10AC	10 Amp Booster Power Supply (ULC)	13 (5.9)
BPS10A/230	10 Amp Booster Power Supply (220V)	13 (5.9)
BPS10CAA	10 Amp Booster Power Supply with California rate	13 (5.9)

1. Requires installation of separate battery cabinet.
2. BPS supports batteries greater than 24 Amp hours for access control applications only.
3. For earthquake anchorage, including detailed mounting weights and center of gravity detail, refer to Seismic Application Guide 3101676. Approval of panel anchorage to site structure may require local AHJ, structural or civil engineer review.

Related Equipment

12V6A5	7.2 Amp Hour Battery, two required	3.4 (1.6)
12V10A	10 Amp Hour Battery, two required	9.5 (4.3)
3-TAMP	Tamper switch	
BC-1EQ	Seismic Kit for BC-1. Order BC-1 separately. See note 3.	
BPSEQ	Seismic kit for BPS6A or BPS10 Booster Power Supplies. See note 3	
BC-1	Battery Cabinet (up to 2 - 40 Amp Hour Batteries)	58 (26.4)
BC-2	Battery Cabinet (up to 2 - 17 Amp Hour Batteries)	19 (8.6)
12V17A	18 Amp Hour Battery, two required (see note 1)	13 (5.9)
12V24A	24 Amp Hour Battery, two required (see note 1)	20 (9.07)
12V40A	40 Amp Hour Battery, two required (see notes 1, 2)	32 (14.5)
12V50A	50 Amp Hour Battery, two required (see notes 1, 2)	40 (18.14)
12V65A	65 Amp Hour Battery, two required (see notes 1, 2)	49 (22.2)

SEALED LEAD ACID **BATTERY, 18AH**

Operations & Maintenance Manual
December 2015

Power Patrol[®]

Batteries

Specification Sheet

Sealed Lead-Acid Batteries

SLA1116

Technical Specifications

Nominal Voltage	12V
Nominal Capacity	18.0 Ah (20 Hr Rate)
Dimensions	Length: 180 mm (7.20 in)
	Width: 76 mm (3.00 in)
	Height: 167 mm (6.60 in)
Total Height w/ Terminal:	167 mm (6.60 in)
Weight	Approx 6.2 Kg (13.85 Lbs.)
Terminal Type	Nut & Bolt (w/faston adapter)

Capacity Characteristics

Cut Off Voltage	20 Hr Rate (0.90A)	18.0AH
1.75 v/c @ 25°C	10 Hr Rate (1.60A)	16.0AH
1.70 v/c	5 Hr Rate (2.9A)	14.5AH
1.55 v/c	1 Hr Rate (9.8A)	9.8AH

Charge Voltage (constant)

	Block	Per Cell
Float	13.5~13.8	2.25~2.30
Cycle	14.4~14.7	2.40~2.45

Discharge Current Amps 250
(5 seconds maximum)

Discharge Current Amps 80
(maximum continuous)

Max. Charge Current 5.1A

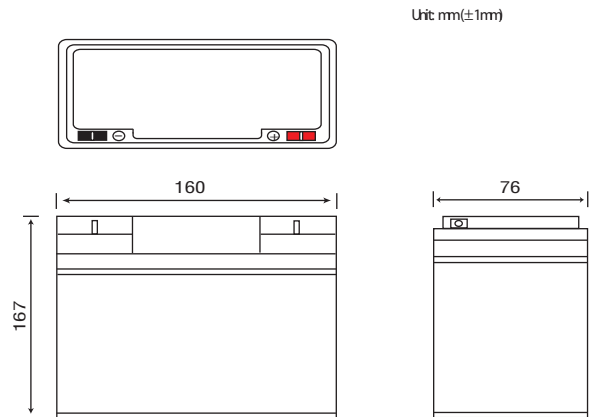
Approx Final Charge Current 0.03 (30 mA)
(2.25 v/c Float)

Approx Final Charge Current 0.15 (150 mA)
(2.45 v/c Cycle)

Self Discharge 9 months@21°C

Case Material ABS - *Gray or Black

Due to changes in the manufacturing process, specifications may change without notice.
* Gray option is Flame Retardant ABS.

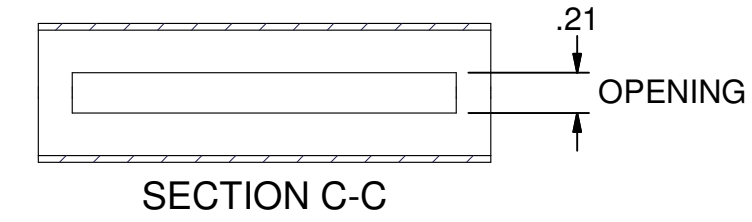
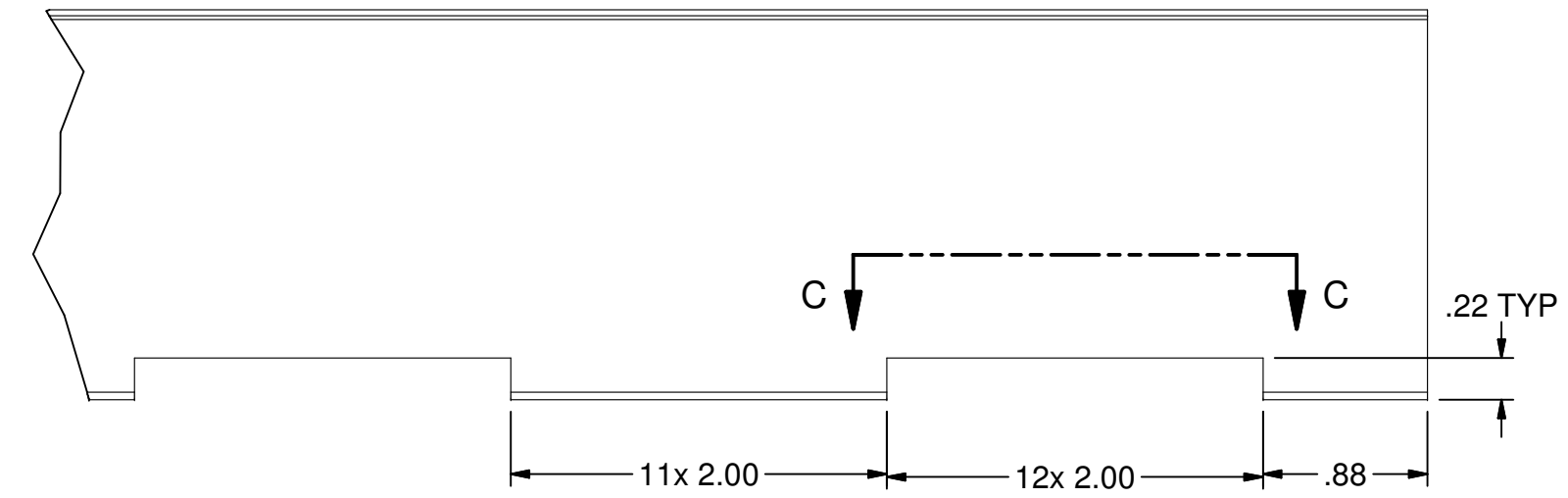
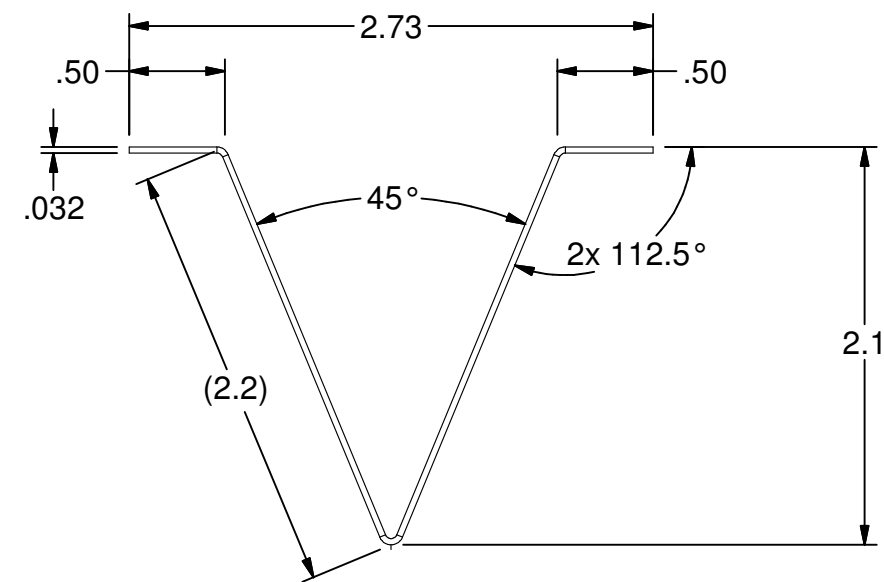
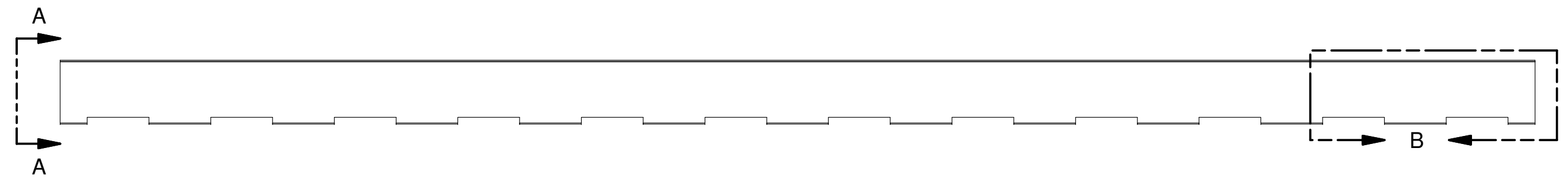
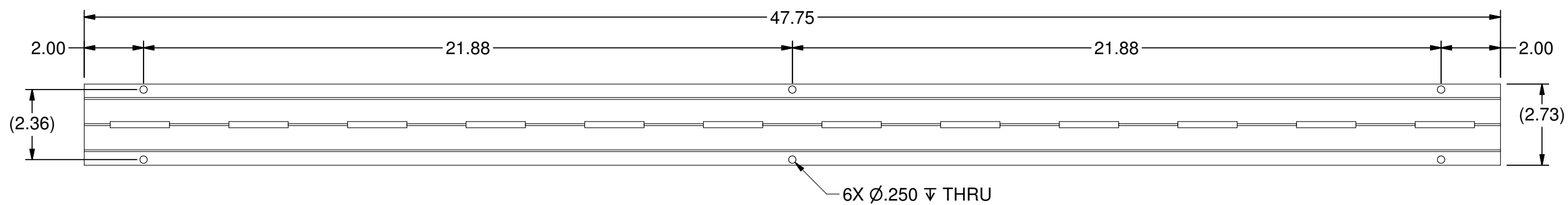


Actual Wattage / Ampere Capacity at Various Discharge Times (Volt per Cell @ 25° C)							
Cut Off Voltage	Time	5 Min	10 Min	15 Min	30 Min	45 Min	60 Min
		1.75 v/c	W	107.4	72.33	55.6	33.79
25°C	A	61.37	41.33	31.77	19.31	14.1	11.3
1.67 v/c	W	104.79	72.16	55.31	33.48	25.05	20.04
25°C	A	62.75	43.21	33.12	20.05	15.0	12.0
1.60 v/c	W	116.8	73.76	54.72	32.19	24.37	19.33
25°C	A	73.0	46.1	34.2	20.12	15.23	12.08

This Page Left Intentionally Blank

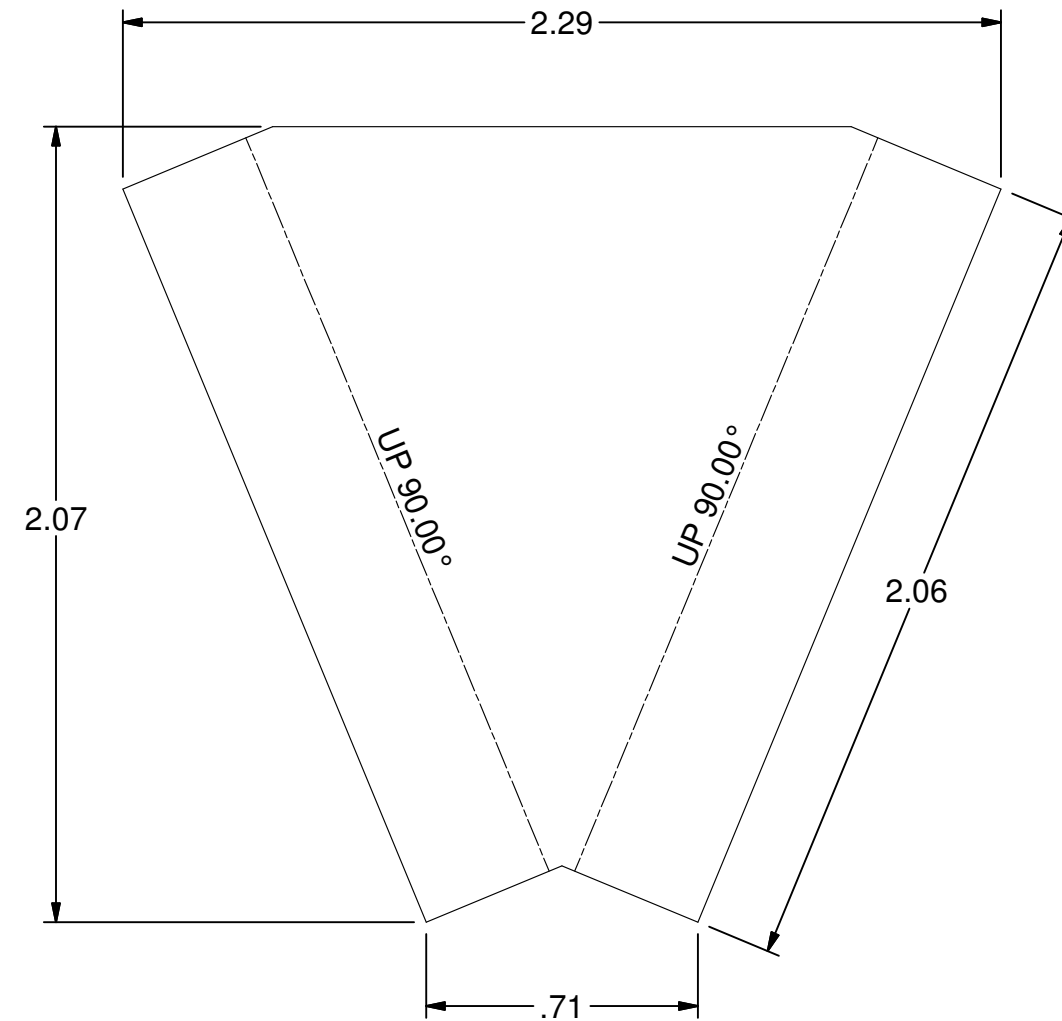
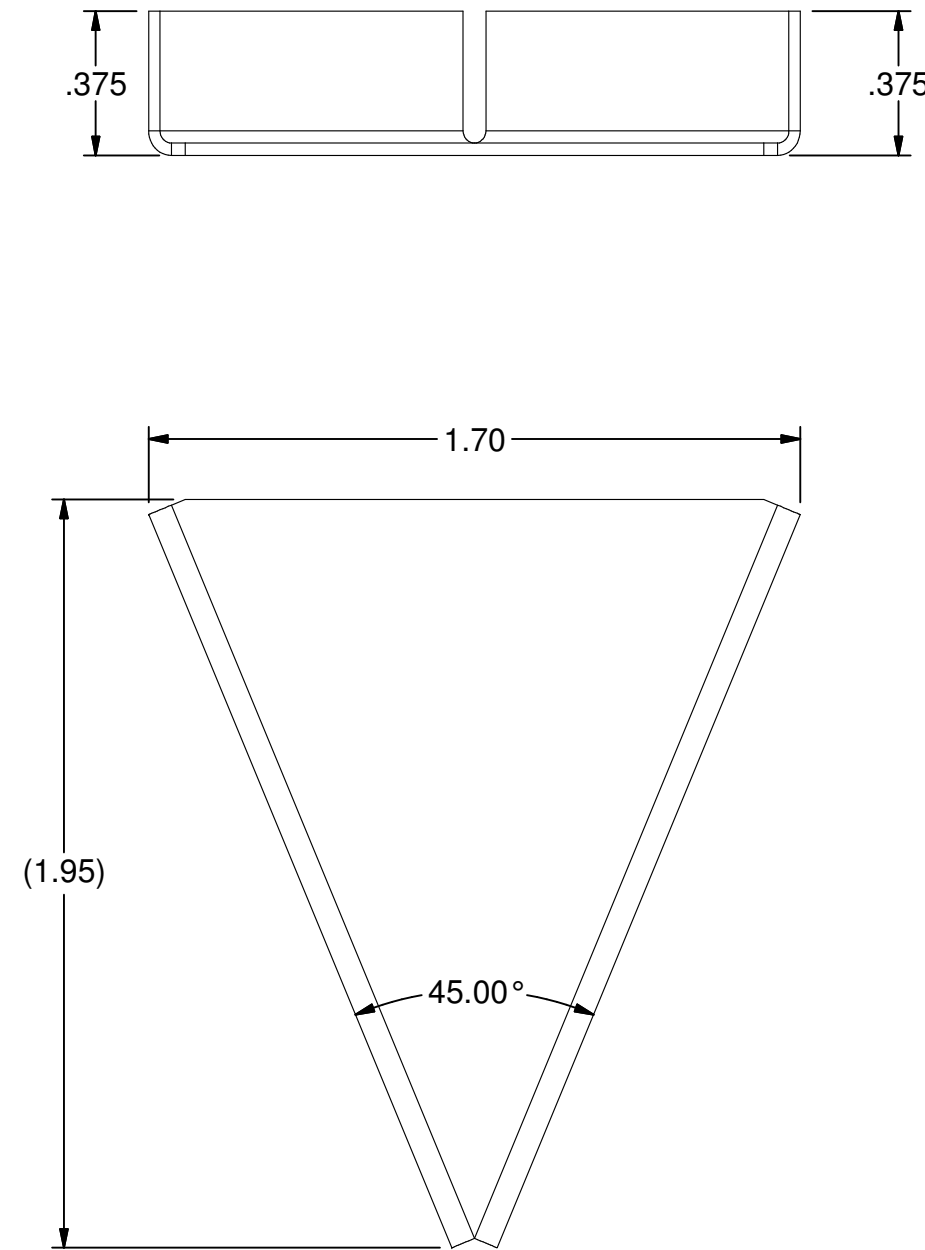
FOLHD HANGER BRACKET

**Operations & Maintenance Manual
December 2015**

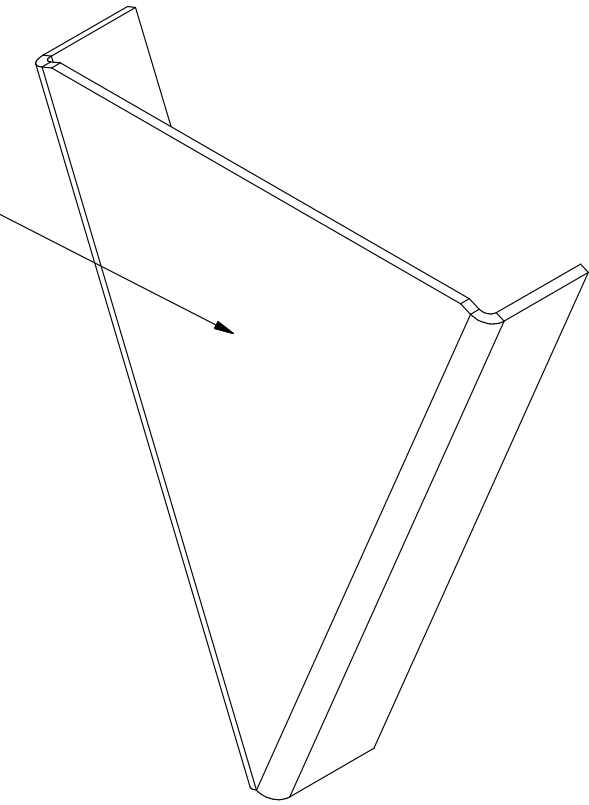


FABRICATE FROM .032 5052 ALUMINUM
 CRYSTAL WHITE PREFINISHED
 (PVC COVERED FOR FABRICATION)

CUSTOMER: SYSTEMS GROUP		PROJECT: EISENHOWER / JOHNSON MEMORIAL TUNNEL	
<p>EASTER-OWENS "IF YOU CAN THINK IT, WE CAN BUILD IT." 6692 Fgo Street Arvado, Colorado 80004 Tel: (303) 431-0111 Fax: (303) 424-2040</p>		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES FRACTIONAL DIMENSIONS: 1/16 DECIMAL DIMENSIONS: .0005 3 PLACE DECIMAL ANGULAR DIMENSIONS: .1°	
		EASTER-OWENS PROJECT NO. 15-29246-1 REVISIONS 3/23/2015 RECORD DRAWING R.P. A	
SCALE: VARIES			
DRAWN BY:	R.P.	3/23/2015	
ENGINEER:	R.P.	3/23/2015	
CHECK BY:	R.P.	3/23/2015	
CAD FILE:	DRAWING.idw		
PART DESCRIPTION: EJMT HANGER			
PART NUMBER: EJMT SUPPORT 45 DEG			
SHEET NUMBER	REV.		
FOI140	- 1.637A		



PAINT SIDE



4 REQUIRED
(MAKE 2 EXTRA FOR SETUP)

END CAP

E DATE: 3/25/2015 **10**
 JOB # 15-29246-1
 R.PHILP
 AUTHORIZED SIGNATURE

CUSTOMER: SYSTEMS GROUP		PROJECT: EISENHOWER / JOHNSON MEMORIAL TUNNEL	
EASTER OWENS "IF YOU CAN THINK IT, WE CAN BUILD IT." 6692 Fiq Street Arvado, Colorado 80004 Tel: (303) 431-0111 Fax: (303) 424-2040		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES FRACTIONAL DIMENSIONS: 1/16 DECIMAL DIMENSIONS: .001 3 PLACE DECIMAL: .000 ANGULAR DIMENSIONS: .1°	
		3RD ANGLE PROJECTION THE PRINT CONTAINS CONFIDENTIAL INFORMATION OWNED EXCLUSIVELY BY EASTER OWENS ELECTRIC, INC. AND IS FURNISHED ON A STRICTLY CONFIDENTIAL BASIS. THIS RESTRICTION INCLUDES, BUT IS NOT LIMITED TO, THE CONDITION THAT THIS PRINT WILL ONLY BE USED AS A RECORD OR TO IDENTIFY OR INSPECT PARTS OR FOR OTHER EQUIPMENT SHOWN IN THIS PRINT BY ANY SOURCE OTHER THAN EASTER OWENS ELECTRIC, INC.	
EASTER-OWENS PROJECT NO. 15-29246-1 REVISIONS 3/25/2015 R.P. RELEASED FOR FABRICATION		SCALE: VARIES DRAWN BY: R.P. 3/25/2015 ENGINEER: R.P. 3/25/2015 CHECK BY: R.P. 3/25/2015 CAD FILE: PRODUCTION DRAWING.idw PART DESCRIPTION: END CAP	
PART NUMBER: ENDCAP		SHEET NUMBER: REV. 101410 - 1.638A	

PRODUCT DATA SHEET**DC 105W-2373**

CUSTOMER:	Nichols Aluminum	
LOCATION:	Decatur, Al	
ATTENTION:	Mike Brougham	
PRODUCT:	19G1240 White	ITEM 126440
CODE:	105W2373	Replaces PPG 582W1118
TYPE:	Polyester	

COATING PROPERTIES

WEIGHT PER GALLON:	10.83 ± 0.20 LBS/GAL
% SOLIDS BY WEIGHT:	65.16 ± 2.00 %
% SOLIDS BY VOLUME:	50.66 ± 2.00 %
PACKAGE VISCOSITY:	24-28 " #4Z @ 78° F OR AS REQUIRED
V.O.C. ; ASTM D 2369	3.77 LBS/GAL
# HAPS PER SOLID GALLON	0.53 LBS PER GALLON
% HAPS BY WEIGHT	2.49 %
PRODUCT SHELF LIFE:	6 months (recheck all properties after 3 months)

APPLICATION PROPERTIES

APPLICATION METHOD:	REVERSE ROLL COAT
APPLICATION VISCOSITY:	14 - 18 " #4Z @ 78° F OR AS REQUIRED
REDUCER:	SC150
SUBSTRATE / GAUGE:	.019 Bare Treated Aluminum
PRETREATMENT:	B1500/B1402W/B1415A/or Equivalent
PRIMER:	N/A
WET FILM THICKNESS:	1.30 - 1.50 MILS
PEAK METAL TEMPERATURE:	435° - 450° F

DRY FILM PROPERTIES

PRIMER AND DRY FILM THICKNESS:	N/A
TOPCOAT DRY FILM THICKNESS:	0.65 - 0.75 MILS
TOTAL DRY FILM THICKNESS:	0.65 - 0.75 MILS
PENCIL HARDNESS:	F - 2H
CURE:	100+ MEK Double Rubs
GLOSS:	85+ @ 60°
T-BEND:	1T NTO ON .019 TREATED ALUMINUM
IMPACT (REVERSE):	28 IN-LB NTO ON .019 ALUM 60"# ON .038 AND ABOVE
ADHESION:	NTO ON 28"# X-HATCHED/IMPACTED AREA
METAL MARKING:	PASS
THEORETICAL COVERAGE:	1160 SQ FT/GAL @ 0.70 MILS
COLOR ON-LINE:	Delta L +or-0.50 Delta a +or-0.50 Delta b +or-0.50

12/10/09

Daniel Shea

DURA COAT PRODUCTS INC.
26655 Peoples Road
Huntsville, Alabama 35756
TEL 256-353-7800
FAX 256-353-9873

PRODUCT DATA SHEET
DC 450C-2000

CUSTOMER: ALERIS
LOCATION: DECATUR, ALABAMA
ATTENTION: Mr. Mike Brougham
PRODUCT: 09E272 H/G CLEAR LAMINATING BACKER
CODE: DC450C2000
TYPE: EPOXY

ITEM 300613

COATING PROPERTIES

WEIGHT PER GALLON:	8.08 ± 0.20 LBS/GAL
% SOLIDS BY WEIGHT:	37 ± 2.00 %
% SOLIDS BY VOLUME:	30 ± 2.00 %
PACKAGE VISCOSITY:	18-22 " #4Z @ 78° F
V.O.C. ; ASTM D 2369	5.01 LBS/GAL
# HAPS PER SOLID GALLON:	6.66 LBS PER SOLID GALLON
% HAPS BY WEIGHT:	24.97%
PRODUCT SHELF LIFE:	6 months (recheck all properties after 3 months)

APPLICATION PROPERTIES

APPLICATION METHOD:	REVERSE ROLL COAT
APPLICATION VISCOSITY:	#4Z @ 78° F AS REQUIRED
REDUCER:	BUTYL CELLOSOLVE/DIACETONE ALCOHOL
SUBSTRATE / GAUGE:	.038 Bare Treated Aluminum
PRETREATMENT:	B1500/B1402W/B1700/or Equivalent
PRIMER:	N/A
WET FILM THICKNESS:	0.50-0.90 MILS
PEAK METAL TEMPERATURE:	435° - 465° F

DRY FILM PROPERTIES

PRIMER AND DRY FILM THICKNESS:	NONE
TOPCOAT DRY FILM THICKNESS:	0.15 - 0.25 MILS
TOTAL DRY FILM THICKNESS:	0.15-0.25 MILS
PENCIL HARDNESS:	H+
CURE:	10+ MEK Double Rubs
GLOSS:	85+ @ 60°
T-BEND:	2T NTO ON .019 TREATED ALUMINUM
IMPACT (REVERSE):	28 IN-LB NTO ON .019 ALUM 60°# ON .038 AND ABOVE
ADHESION:	NTO ON 28°# X-HATCHED/IMPACTED AREA
METAL MARKING:	PASS
THEORETICAL COVERAGE:	2386 SQ FT/GAL @ 0.20 MILS
COLOR ON-LINE:	n/a

WAX FREE FOR LAMINATING AND BONDING

*MAY RUN DC477, DC425Y2000 OR 425Y15A PRIMER UNDER THIS PRODUCT FOR APPROVED APPLICATIONS

**MAY RUN AT 0.30 MILS DFT DEPENDING ON END CUSTOMER SPECIFICATIONS

04/30/2014

Dexter F. Sunderman

M A T E R I A L S A F E T Y D A T A S H E E T

19G1240 WHITE

Page: 1
12/10/2009

PRODUCT NAME: 19G1240 WHITE
PRODUCT CODE: DC105W-2373

HMIS CODES: H F R P
1 2 0 G

===== SECTION I - MANUFACTURER IDENTIFICATION =====

MANUFACTURER'S NAME: DURA COAT PRODUCTS OF AL, INC.
ADDRESS : 26655 PEOPLES ROAD
HUNTSVILLE, AL 35756

EMERGENCY PHONE : 1-800-424-9300
INFORMATION PHONE : 1-256-353-7800
DATE PRINTED : 12/10/2009
NAME OF PREPARER : Dexter F. Sunderman
Dura Coat Products' Customer:

===== SECTION II - HAZARDOUS INGREDIENTS/SARA III INFORMATION =====

REPORTABLE COMPONENTS	CAS NUMBER	VAPOR PRESSURE		WEIGHT
		mm Hg @ TEMP		PERCENT
TITANIUM DIOXIDE	13463-67-7	NA	NA	28.566
TLV ACGIH 10 MG/M3 PEL OSHA 15 MG/M3				
POLYESTER RESIN	P TRADE SECR	3.02	68	21.140
AROMATIC HYDROCARBON	64742-94-5	0.50	68	12.055
TLV ACGIH 100 PPM PEL OSHA 100 PPM				
POLYESTER RESIN	PT SECRET	3.02	68	7.313
HFSN AROMATIC HYDROCARBON	64742-95-6	2.09	68	6.475
TLV ACGIH 100 PPM PEL OSHA 100 PPM				
METHYLATED MELAMINE	068002-20-0	NA	NA	5.057
* ETHYLENE GLYCOL MONOBUTYL ETHER	111-76-2	.88	77	4.327
TLV ACGIH 20 PPM PEL OSHA 25 PPM				
* 1,2,4-TRIMETHYLBENZENE/PSEUDOCUEME	95-63-6	1.7	68	4.112
TLV ACGIH 25 PPM PEL OSHA 25 PPM				
DIACETONE ALCOHOL	123-42-2	1	68	2.047
TLV ACGIH 50 PPM PEL OSHA 50 PPM				
MODIFIED POLYESTER RESIN	TRD SECRET	2.6	68	1.800
PROPYLENE GLYCOL METHYL ETHER ACETATE	108-65-6	3.7	68	1.239
TLV ACGIH 100 PPM PEL OSHA 100 PPM				
*# XYLENE AROMATIC HYDROCARBON	1330-20-7	14	100	1.237
TLV ACGIH 100 PPM PEL OSHA 100 PPM				
*# NAPHTHALENE	91-20-3	NA	NA	1.123
TLV ACGIH 10 PPM PEL OSHA 10 PPM				
FUMED SILICA	112945-52-5	NA	NA	.440
TLV ACGIH 10MG/M3 PEL OSHA 6 MG/M3				
ISOPROPYL ALCOHOL	67-63-0	32.48	68	.305
TLV ACGIH 200 PPM PEL OSHA 400 PPM				
*# CHROME ANTIMONY TITANATE	68186-90-3	N/A	N/A	.148
*CHROMIUM III COMPOUNDS 5%				
*ANTIMONY COMPOUNDS 12%				
*# ISOPROPYLBENZENE/CUMENE	98-82-8	NA	NA	.143
TLV ACGIH 50 PPM PEL OSHA 50 PPM				
*# ETHYLBENZENE	100-41-4	14.2	100	.122
TLV ACGIH 100 PPM PEL OSHA 100 PPM				
ISOBUTYL ALCOHOL	78-83-1	10	86	.106
TLV ACGIH 50 PPM PEL OSHA 100 PPM				
SULFONIC ACID CATALYST	104-15-4	NA	NA	.102

M A T E R I A L S A F E T Y D A T A S H E E T

19G1240 WHITE

Page: 2
12/10/2009

* 2-PROPOXYETHANOL	2807-30-9	1.3	68	.075
*# FORMALDEHYDE	50-00-0	10	86	.037
TLV ACGIH 0.3 PPM PEL OSHA 0.75PPM				
2ETHYL-1HEXANOL/OCTYL ALCOHOL	104-76-7	.1	86	.005
TLV ACGIH 227 PPM PEL OSHA 227 PPM				
PIGMENT BLACK 6	1333-86-4	NA	NA	.003
TLV ACGIH 3.5 MG/M3 PEL OSHA 3.5 MG/M3				
PIGMENT RED 101	1309-37-1	N/A	N/A	.001
SILICON DIOXIDE (7631-86-9) 3.00% TLV ACGIH 10 MG/M3 PEL (OSHA) 80 mg/m3				
BARIUM SULFATE CAS#7727-43-7 5.00%				

* Indicates toxic chemical(s) that's subject to the reporting requirements of section 313 of Title III and 40 CFR 372.
Indicates chemical(s) subject to the reporting requirements of the U.S. Hazardous Air Pollutants Act.

WARNING: Detectable amounts of a chemical known to the state of California to cause cancer, birth defects or other reproductive harm may be present in this product.

===== SECTION III - PHYSICAL/CHEMICAL CHARACTERISTICS =====

BOILING RANGE: 101 deg F - 364 deg F	SOLUBILITY IN WATER: Non Soluble
VAPOR DENSITY: Heavier than air	EVAPORATION RATE: Slower than ether.
WEIGHT PER GALLON (LBS/GAL): 10.833	SPECIFIC GRAVITY (H2O=1): 1.3
WEIGHT SOLIDS %: 65.158	VOLATILE WEIGHT %: 34.842
V.O.C. (LBS/GAL): 3.77	COATING VOLATILE CONTENT: 5.23 lb/g1
APPEARANCE AND ODOR: Liquid with an aromatic odor.	

===== SECTION IV - FIRE AND EXPLOSION HAZARD DATA =====

FLASH POINT OF SYSTEM: 105 deg F	METHOD USED: TCC
FLASH POINT LOWEST FLASHING SOLVENT: 55 deg F	METHOD USED: TCC
FLAMMABLE LIMITS IN AIR BY VOLUME- LOWER: 0.9	UPPER: 15.8

EXTINGUISHING MEDIA:

Use National Fire Protection Association (NFPA) CLASS B Extinguishers (Foam, Alcohol Foam, CO2, or Dry Chemical), designed to extinguish NFPA CLASS II Combustible Liquid fires. Water spray may be ineffective. Water spray/fog may be used to cool closed containers to prevent pressure build-up and possible auto-ignition or explosion when exposed to extreme heat.

SPECIAL FIREFIGHTING PROCEDURES

Respiratory equipment should be worn to avoid inhalation of vapors. Water should not be used except as fog to keep nearby containers cool.

UNUSUAL FIRE AND EXPLOSION HAZARDS

Handle as flammable liquid. Vapors form an explosive mixture in air between the upper and lower explosive limits which can be ignited by many sources such as pilot lights, open flames, electrical motors and switches.

===== SECTION V - REACTIVITY DATA =====

STABILITY:

Stable.

CONDITIONS TO AVOID

Excessive heat, poor ventilation, corrosive atmospheres, excessive aging.

INCOMPATIBILITY (MATERIALS TO AVOID)

Alkaline materials, strong acids and oxidizing materials.

HAZARDOUS DECOMPOSITION OR BYPRODUCTS

Carbon monoxide, carbon dioxide, oxides of nitrogen, and possibly lower molecular weight fractions .

HAZARDOUS POLYMERIZATION:

Will not occur.

===== SECTION VI - HEALTH HAZARD DATA =====

INHALATION HEALTH RISKS AND SYMPTOMS OF EXPOSURE

Dizziness, breathing difficulty, headaches & loss of coordination.

SKIN AND EYE CONTACT HEALTH RISKS AND SYMPTOMS OF EXPOSURE

Eye contact: Severe irritation, tearing, redness and blurred vision.

SKIN ABSORPTION HEALTH RISKS AND SYMPTOMS OF EXPOSURE

Skin contact: Can dry and defat skin causing cracks, irritation, and dermatitis.

INGESTION HEALTH RISKS AND SYMPTOMS OF EXPOSURE

Can cause gastrointestinal irritation, vomiting, nausea, and diarrhea.

HEALTH HAZARDS (ACUTE AND CHRONIC)

Inhalation-Dizziness, breathing difficulty, headaches, & loss of coordination. Eye contact-Severe irritation, tearing, redness, and blurred vision. Skin contact-Can dry and defat skin causing cracks, irritation, and dermatitis. Ingestion-Can cause gastrointestinal irritation, vomiting, nausea, & diarrhea. No chronic health effects. Additional hazardous info.

CARCINOGENICITY:

NTP CARCINOGEN: Yes IARC MONOGRAPHS: Yes OSHA REGULATED: No

MEDICAL CONDITIONS GENERALLY AGGRAVATED BY EXPOSURE

Anesthesia, respiratory tract irritation, dermatitis, nausea, vomiting

EMERGENCY AND FIRST AID PROCEDURES

Inhalation overexposure-Move person to fresh air. If breathing stops, apply artificial respiration and seek immediate medical attention. Eye contact-flush with large quantities of water for 15 minutes. Skin contact-Wash thoroughly with soap and water and see a doctor. Ingestion-Do not induce vomiting, can cause chemical pneumonitis and pulmonary edema. Contact physician immediately.

===== SECTION VII - PRECAUTIONS FOR SAFE HANDLING AND USE =====

STEPS TO BE TAKEN IN CASE MATERIAL IS RELEASED OR SPILLED

Eliminate ignition sources, provide good ventilation, dike spill area and add absorbent earth or sawdust to liquid. Thoroughly wet w/ water and mix.

WASTE DISPOSAL METHOD

Collect absorbent/water/spilled liquid mixture into metal containers and add enough water to cover. Consult local, state & federal hazardous waste regulat'n before disposing into approved hazardous waste landfills. Obey relevant laws.

PRECAUTIONS TO BE TAKEN IN HANDLING AND STORING

Use non-sparking utensils when handling this material. Avoid hot metal surface. Avoid temperature extremes during storage. Use in cool, well-ventilated areas. Keep containers closed when not in use. Keep away from excessive heat, sources of ignition, and from reactive materials. Material can burn; limit indoor storage to approved area equipped with automatic sprinklers. Avoid all ignition sources. Ground all metal containers during storage and handling. Use non-sparking utensils when handling this material. Avoid hot metal surface. Use in cool, well-ventilated areas. Keep containers closed when not in use. Keep away from excessive heat and open flames. Avoid temperature extremes during storage. Store away from excessive heat, from sources of ignition and from reactive materials. Store in a dry area. Ground all metal containers during storage and handling. Material can burn; limit indoor storage to approved areas equipped with automatic sprinklers. Avoid all ignition sources.

OTHER PRECAUTIONS

Smoking in area where this material is used should be strictly prohibited. Tools used with this material should be made from aluminum, brass or copper. Plastic utensils should not be used. NOTE: This information is accurate to the best knowledge of Dura Coat Products Inc., but is furnished without any expressed or implied warranties.

===== SECTION VIII - CONTROL MEASURES =====

RESPIRATORY PROTECTION

When spraying this material use a NIOSH approved cartridge respirator or gas mask suitable to keep airborne mists and vapor concentrations below the time weighted threshold limit values. When using in poorly ventilated and confined spaces, use a fresh-air supplying respirator or a self-contained breathing apparatus.

VENTILATION

General mechanical ventilation or local exhaust should be suitable to keep vapor concentrations below TLV. Ventilation equip. must be explosion proof.

PROTECTIVE GLOVES

Impermeable chemical handling gloves for skin protection. A NIOSH approved full-facedpiece, air-purifying respirator, or full facepiece, airline respirator in the pressure demand should be worn when working with this material. Use chemical resistant gloves should be worn whenever this material is handled.

EYE PROTECTION

Wear Safety Goggles and or full face shield.

OTHER PROTECTIVE CLOTHING OR EQUIPMENT

Use impermeable aprons and protective clothing whenever possible to prevent skin contact. The use of head caps whenever possible is strongly recommended.

WORK/HYGIENIC PRACTICES

Eye washes and safety showers in the workplace are recommended.

===== SECTION IX - DISCLAIMER =====

To the best of our knowledge, the information contained herein is accurate, obtained from sources believed by Dura Coat Product's Inc. to be accurate.

PART NO.

Att:

NICHOLS ALUMINUM-NAA
CERTIFIED INSPECTION REPORT

To: O'NEAL FLAT ROLLED METALS
1229 SOUTH FULTON AVE
BRIGHTON, CO 80601
USA

Date: 11/26/14
Ship Date: 12/01/14

Nichols Sales Order Number: 117506
Customer Purchase Order No.: BR-24266
Nichols Part No.: 124621

Line Number: 2
Customer Part No.: NA124621
Bill of Lading No.: NAA43108
Total Net Weight: 26,082.0

Description: .0290X49.000 3105 H24 WHITE(1118)/CLEAR

CHEMICAL INFORMATION:

Coil #	SI	FE	CU	MN	MG	CR	ZN	TI
590245	0.32	0.69	0.17	0.48	0.53	0.08	0.15	0.03
590247	0.31	0.65	0.17	0.47	0.52	0.08	0.15	0.03

MECHANICAL INFORMATION:

Coil #	H/T	UTS (KSI)	YTS (KSI)	%EL	
590245	H	25.5	21.2	11	Actual Gauge = 0.0284
590247	H	26.3	23.3	11	Actual Gauge = 0.0287

Report approved by Nichols Aluminum Quality Manager

This product conforms to the applicable paragraphs of ASTM B209 (latest revision).

MATERIAL WAS MELTED, ROLLED AND PROCESSED IN THE USA

This Page Left Intentionally Blank

Aleris
Rolled Products

t.15
t.02
t.01

Aleris Code C4-18427 / 19G1240 Submit Date _____

Color Name White ARP _____

Paint System Polyester Track # _____

Approved By Paul Toff Date 4/6/15

Aleris
Rolled Products

This Page Left Intentionally Blank

FOLHD HANGER FASTENERS

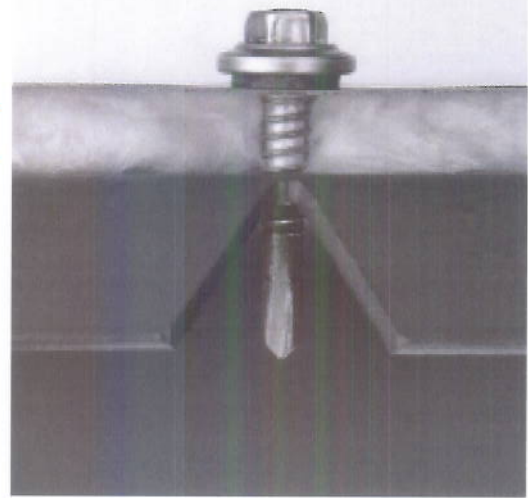
**Operations & Maintenance Manual
December 2015**

SCOTS®

Long Life Tek® Fasteners

The Original Buildex Scots 300 Stainless Steel Encapsulated

Head will never rust EVER!!!



Applications

- Panel over blanket insulation to steel frame.
- Roof curb to structural steel.
- Standing seam roof system endlaps.
- Stitch roof and wall sidelaps.
- #17 diameter replacement screw for #12 and 1/4 diameter fasteners.

Product Features

- For improved appearance and extra life an integral design controls the EPDM rubber washer.
- Combines 300 series stainless steel exterior with the strength of carbon steel.
- 300 series stainless most preferred for galvalume long life panels.
- Maximum sealing and corrosion protection.
- Point has precise cutting edges to improve drill performance with less effort.
- Non-walking point provides fast material engagement.
- Point to thread design maximizes pullout performance and minimizes backout.
- Drills and taps in the broadest range of applications.

Product Specifications

Diameter.....	#12, 1/4 and #17
Thread Form.....	12-14 12-24 1/4-14 17-14
Head Style.....	5/16" HWH (300 Series Stainless)
Washer Style.....	9/16" O.D. Integral
Drill Point.....	Teks 1 Teks 2 Teks 3 Teks 5 Type AB
Finish.....	Climaseal™ Available with P ³ powder paint

Head Style



Scots Head



Copper State Bolt & Nut Co.
 13365 James E. Casey Ave.
 Englewood, CO 80112
 (303) 799-1804 Fax (303) 799-6154

Selector Guide



Part Number	Description	Head Style	Drill Point	Drill & Tap Capacity	Max. Material Attachment	Box Qty	Applications
1165253	12-14 x 1"	HWH	#3	.036-.210	.310	2,500	• Panel over blanket insulation to steel frame. • Roof curb to structural steel.
1167253	12-14 x 1-1/4"	HWH	#2	.036-.210	.560	2,500	
1168253	12-14 x 1-1/2"	HWH	#2	.036-.210	.790	2,000	
1163000	12-24 x 2"	HWH	#5	.125-.500	1.000	1,500	
1387253	1/4-14 x 7/8"	HWH	#1	.018-.095	.320	2,500	
1175153	1/4-14 x 1-1/8"	HWH	#1	.018-.095	.395	2,500	• Standing seam roof system endlaps.
1616000	1/4-14 x 1-1/4"	HWH	#3	.036-.210	.630	2,000	• Stitch roof and wall sidelaps. SSR endlaps.
1178253	1/4-14 x 3/4"	HWH	AB	.012-.048	.435	2,500	• Replacement screw for #12 and 1/4 diameter fasteners.
1180225	17-14 x 1"	HWH	AB	-	.686	1,500	

Performance Data

PULLOUT VALUES (average lbs. ultimate)											
Fastener		Steel Gauge									
Dia.	Pt.	26	24	22	20	18	16	14	12	3/16	1/4
12	2	156	243	283	375	605	848	1181	1856	3520	-
12	5	-	-	-	-	-	-	-	1527	3701	3999
1/4	5	-	-	-	-	-	-	-	1507	3300	5059
1/4	1	208	329	428	562	800	1151	-	-	-	-
	AB	235	365	496	597	702	926	1180	1690	3446	4212
17	AB	225	261	350	442	553	752	1122	1716	3523	4250

FASTENER VALUES			
Fastener (dia-tpi)	Tensile (lbs. min.)	Shear (avg. lbs. ult.)	Torque (min. in. lbs.)
12-14	2778	2000	92
12-24	3188	2100	100
1/4-14	4060	2600	150
1/4-28	5577	3310	234
17-14	5300	3200	180

SHEET STEEL GAUGES								
Gauge No.	12	14	16	18	20	22	24	26
Decimal Equivalent	.105"	.075"	.060"	.048"	.036"	.030"	.024"	.018"

SHEAR VALUES (average lbs. ultimate)											
Fastener		Steel Gauge (lapped)									
Dia.	Pt.	26	24	22	20	18	16	14	1/8	3/16	1/4
12	2	365	600	623	898	1370	1758	2138	2202	-	-
12	5	-	-	-	-	-	-	-	2700	2720	2762
1/4	5	-	-	-	-	-	1597	2005	2350	2792	3310
1/4	1	511	849	885	1244	1764	-	-	-	-	-
	AB	422	482	669	889	1157	-	-	-	-	-
17	AB	459	745	989	958	1415	-	-	-	-	-

The values listed are ultimate averages achieved under laboratory conditions and apply to Buildex manufactured fasteners only. Appropriate safety factors should be applied to these values for design purposes.

Installation Guidelines

- A standard screwgun with a depth sensitive nosepiece should be used to install Tek's®. For optimal fastener performance, the screwgun should be a minimum of 6 amps and have an RPM range of 0-2500.
- Adjust the screwgun nosepiece to properly seat the fastener.
- New magnetic sockets must be correctly set before use. Remove chip build-up as needed.
- Overdriving may result in torsional failure of the fastener or stripout of the substrate.
- The fastener must penetrate beyond the metal structure a minimum of 3 pitches of thread.

Properly Seated Washers



1349 West Bryn Mawr Avenue
Itasca, Illinois 60143
630-595-3500 Fax: 630-595-3549
www.itwbuildex.com

Scots®, Tek's®, ABOT™, and Climaseal® are trademarks of ITW Buildex and Illinois Tool Works, Inc.

© 2010 ITW Buildex and Illinois Tool Works, Inc.

This Page Left Intentionally Blank

FIBER-OPTIC LINEAR
HEAT DETECTION
(FOLHD) PARTS LIST
SECTION

Operations & Maintenance Manual
December 2015

Eisenhower Johnson Memorial Tunnel - EMJT
 Fixed Fire Suppression System Project
 Parts List

Part #	Description	Qty
LHD3-04	LIOS Optical Linear Heat Detector, 2-Channel, 4km	2
Charon3	LIOS Optical Linear Heat Detector Software	2
28024033	LIOS Splice Box with Grounding System, 4-Channel	4
28024036	LIOS Termination Box with Grounding System	2
28030504	LIOS Sensor Cable, Fast Response, FRNC, 4.0mm (Tunnel)	20,000 ft
28030606	LIOS Sensor Cable, Fast Response, FRNC 4.0mm (Non-Tunnel)	2,000 ft
28024020	LIOS Single Fiber Cable Pigtail, E2000 APC Connector, 62.5m	4
759-911	WAGO System 750 I/O Pro Software	1
750-881	WAGO 750-881 Ethernet Programmable Fieldbus Controller	2
787-1012	WAGO 787-1012 Switched Mode Power Supply	14
750-513	WAGO 750-513 2-Channel Relay Output Module	254
750-613	WAGO 750-613 Internal System Power Supply Module	8
750-627	WAGO 750-627 Internal Data Bus Extension End Module	4
750-628	WAGO Internal Data Bus Extension Coupler Module	2
750-600	WAGO End Module	8
BPS10A	Remote Booster Power Supply, 10 Amp (LIOS Detector)	2
SLA1116	Sealed lead Acid Battery, 18AH	4
N/A	FOLHD Hanger Bracket	4,500
N/A	FOLHD Hanger Bracket End Cap	4

FIBER-OPTIC LINEAR
HEAT DETECTION
(FOLHD)
RECOMMENDED SPARE
PARTS LIST
SECTION

Operations & Maintenance Manual
December 2015

Eisenhower Johnson Memorial Tunnel - EMJT
 Fixed Fire Suppression System Project
 Recommended Spare Parts List

Part #	Description	Qty
LHD3-04	LIOS Optical Linear Heat Detector, 2-Channel, 4km	0
Charon3	LIOS Optical Linear Heat Detector Software	0
28024033	LIOS Splice Box with Grounding System, 4-Channel	0
28024036	LIOS Termination Box with Grounding System	0
28030504	LIOS Sensor Cable, Fast Response, FRNC, 4.0mm (Tunnel)	500 ft
28030606	LIOS Sensor Cable, Fast Response, FRNC 4.0mm (Non-Tunnel)	500 ft
28024020	LIOS Single Fiber Cable Pigtail, E2000 APC Connector, 62.5m	2
759-911	WAGO System 750 I/O Pro Software	0
750-881	WAGO 750-881 Ethernet Programmable Fieldbus Controller	1
787-1012	WAGO 787-1012 Switched Mode Power Supply	2
750-513	WAGO 750-513 2-Channel Relay Output Module	10
750-613	WAGO 750-613 Internal System Power Supply Module	2
750-627	WAGO 750-627 Internal Data Bus Extension End Module	1
750-628	WAGO Internal Data Bus Extension Coupler Module	1
750-600	WAGO End Module	1
BPS10A	Remote Booster Power Supply, 10 Amp (LIOS Detector)	0
SLA1116	Sealed lead Acid Battery, 18AH	0
N/A	FOLHD Hanger Bracket	50
N/A	FOLHD Hanger Bracket End Cap	2

FIBER-OPTIC LINEAR
HEAT DETECTION
(FOLHD)
EQUIPMENT & SERVICE
PROVIDERS

Operations & Maintenance Manual
December 2015

Eisenhower Johnson Memorial Tunnel
Fixed Fire Suppression System
FOLHD Equipment & Service Providers

Local Equipment & Service Provider

Systems Group
800 East 64th Avenue, Unit 17
Denver, CO 80229

FOLHD Hanger Bracket/End Cap

Easter Owens
6692 Fig Street
Arvada, CO 80004

Linear Heat Detection Equipment

LIOS Technology, Inc.
1400 Campus Drive West
Morganville, NJ 07751

WAGO I/O System Equipment

LIOS Technology, Inc.
1400 Campus Drive West
Morganville, NJ 07751

FIBER-OPTIC LINEAR
HEAT DETECTION
(FOLHD)
CONSUMABLE SUPPLIES
SECTION

Operations & Maintenance Manual
December 2015

Eisenhower Johnson Memorial Tunnel
Fixed Fire Suppression System
FOLHD System

Consumable Supplies

The FOLHD System and all associated components have no required consumables for normal operations, nor for any ongoing testing and maintenance operations.

FIBER-OPTIC LINEAR
HEAT DETECTION
(FOLHD)
TESTING &
COMMISSIONING
SECTION

Operations & Maintenance Manual
December 2015

Eisenhower Johnson Memorial Tunnel
Fixed Fire Suppression System
FOLHD System

Testing & Maintenance Requirements

The LIOS FOLHD detector is essentially maintenance free; however, some yearly operational tests should be performed. These yearly tests include:

- Check all sensor cables and pigtails for visible damages such as breaks, cuts, kinks or tight bending
- Switch off the controller and wait for at least 5 seconds
- Switch on the controller and check if all front LEDs lit for a few seconds
- Check if the SYSTEM READY LED lits within 30 seconds
- Connect the controller to a computer and start QUICK START function of the Charon3 software
- Select the controller and scan all channels
- Check the result of the QUICK START for failures and changes compared to the previous QUICK START
- Perform application specific tests as described in the Operation Manuals and local guidelines
- Take appropriate measures in case of failures

Specific servicing and testing recommendations can be found in Chapter 6 Section 6.2.3 of the LIOS Operations Manual, included in this O&M.

Additional required testing of the FOLHD System, as a function of the overall fixed fire suppression system is addressed in the Short Term Operation Plan (STOP) and the Annual Maintenance Plan (AMP), which are provided under separate submittal.