



WAGO SPEEDWAY 767

ETHERNET, 8DI 24 V DC

767-2301

Programmable Fieldbus Coupler

Version 2.1.1

Introduction

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. Please feel free to e-mail us at documentation@wago.com.

Service and Technical Support

Additional information regarding this and other products (e.g., data sheets) is available on our website: www.wago.com.

If you can not eliminate errors or faults using the measures described in this manual we will be glad to assist you further. Contact us at:

AUTOMATION Support

Phone: +49 571 887 555

Fax: +49 571 887 8555

E-mail: support@wago.com

Additional Support

The Seminar and Training department offers useful seminars to support you in the use of WAGO products. For more information, visit our website or call +49 571 887-327, or send an e-mail to training@wago.com.

Trademarks

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in this manual are generally protected by trademark or patent.

Copyright © 2011 by WAGO Kontakttechnik GmbH & Co. KG. All rights reserved.

Table of Contents

Introduction	2
1 Notes about these Operating Instructions	9
1.1 Validity of these Operating Instructions	9
1.2 Explanation of Symbols	10
2 Safety	12
2.1 Use in Compliance with Underlying Provisions	12
2.2 Personnel Qualification	12
2.3 Basic Safety Information	13
2.4 Safety Equipment	14
2.5 Technical Condition of the 767 Series components	15
2.6 Notes on Operation	15
3 Information on Ethernet	16
3.1 Physical Transmission into Network	17
3.1.1 Transfer rates	17
3.1.2 Network Topology	18
3.1.3 Transmission modes	19
3.1.4 Ethernet hardware address (MAC-ID)	21
3.2 Logical Addressing	22
3.2.1 IP – Internet Protocol	22
3.2.2 IP addresses	23
3.2.3 Subnet	25
3.2.4 Gateway	26
3.2.5 RAW IP	26
3.2.6 IP Multicast	27
3.3 Transport protocols	28
3.4 Configuration and Diagnostic Protocols	30
4 Information on EtherNet/IP	34
5 Information on MODBUS	37
6 Fieldbus Coupler Description	38
6.1 Connections	39
6.2 Marking Possibilities and Fasteners	40
6.3 LEDs and Operating Elements	41
6.4 Address and Operating Mode Switch (DIP Switch)	43
6.5 Labeling and Symbols on the Back Side	44
6.6 Lateral Marking	45
6.7 Block diagram	46
6.8 Dimensions	47
6.9 Technical Data	48

7	Mounting the Fieldbus Coupler.....	50
7.1	Information on Mounting.....	50
7.2	Tools and Accessories Required for Mounting.....	51
7.3	Direct Mounting on Your System.....	52
7.4	Mounting on a Carrier Rail (only with WAGO Accessories).....	53
7.4.1	Fastening the Carrier Rail Adapter to the Fieldbus Coupler.....	53
7.4.2	Fastening the Fieldbus Coupler with Carrier Rail Adapter to a Carrier Rail.....	54
7.5	Mounting on a Profile Rail (only with WAGO Accessories).....	55
7.5.1	Fastening the Profile Adapter to the Fieldbus Coupler.....	55
7.5.2	Fastening the Fieldbus Coupler with Profile Adapter to a Profile Rail.....	56
7.6	Replacing the Marking Spaces.....	57
7.7	Mounting the Spacer in the Case of Compact Arrangement.....	58
8	Connection to the Data and Supply Cable.....	60
8.1	Notes.....	60
8.2	Required Accessories.....	61
8.3	Connecting the Ethernet Cable.....	62
8.3.1	Connecting a fieldbus coupler to an Ethernet network.....	63
8.3.2	Connecting several fieldbus couplers inside an Ethernet network.....	64
8.4	Connecting the S-BUS Cable.....	66
8.5	Connecting the Supply Cable.....	68
8.6	Connecting the Sensor Cable.....	70
8.7	Connecting the USB Cable.....	72
9	Start-Up	74
9.1	Determining the IP Address of the Host PC.....	75
9.2	Assignment of an IP Address.....	76
9.3	Assignment an IP Address using BOOTP.....	77
9.4	Setting a Permanent IP Address using DIP Switches.....	80
9.4.1	Assigning an IP Address using the Web-Based Management.....	82
9.4.2	Assigning an IP Address using WAGOframe.....	82
9.5	Testing the Network Connection.....	83
9.6	Switching on the Fieldbus Coupler.....	84
10	Configuration	85
10.1	Access Configuration of Output Modules.....	85
10.2	The Web-Based Management (WBM).....	86
10.2.1	"Information" View.....	87
10.2.2	"TCP/IP" View.....	89
10.2.3	"Port" View.....	91
10.2.4	"Watchdog" View.....	92
10.2.5	"Clock" View.....	95
10.2.6	"Ethernet" View.....	96
10.2.7	"Users" View.....	98
10.2.8	"Administration" View.....	99
11	File System.....	100
11.1	User Administration.....	100
11.2	Access via FTP.....	101

12	EtherNet/IP.....	102
12.1	EDS File	102
12.2	Process Image.....	103
12.3	Process Data Exchange	104
12.3.1	Assembly Instances	104
12.3.2	Identifying the Data Length.....	106
12.3.3	Example with Assembly Instances 101 and 104	107
12.4	Diagnostics	109
12.4.1	Diagnostics Integrated in the Input and Output Data	110
12.4.2	Diagnostics via Explicit Messages	113
12.4.3	Diagnostics Mixed via Input and Output Data and Explicit Messages	114
12.5	Object Model.....	115
12.6	CIP Classes.....	116
12.6.1	Identity Object (01 _{hex})	118
12.6.2	Message Router Object (02 _{hex})	120
12.6.3	Assembly Object (04 _{hex})	122
12.6.4	Connection Object (05 _{hex})	129
12.6.5	Connection Manager Object (06 _{hex}).....	129
12.6.6	Port Class Object (F4 _{hex}).....	130
12.6.7	TCP/IP Interface Object (F5 _{hex}).....	132
12.6.8	Ethernet Link Object (F6 _{hex})	134
12.7	WAGO-Specific Classes	139
13	MODBUS.....	141
13.1	Process Image.....	143
13.2	Process Data Exchange	143
13.3	Accessing the Process Image via MODBUS Functions	144
13.3.1	Register Services	145
13.3.2	Bit Services.....	147
13.3.3	Configuration Register	149
13.3.4	Watchdog Behavior	150
13.3.4.1	Watchdog Register.....	151
13.3.4.2	Diagnostics Registers.....	154
13.3.4.3	Configuration Registers	154
13.3.4.4	Firmware Information Register	157
13.3.4.5	Constant Register	159

14	CoDeSys 3 Runtime Environment.....	161
14.1	Installing the CoDeSys Programming System.....	161
14.2	Editing the Gateway Configuration for CoDeSys 3.....	162
14.3	Programming with CoDeSys 3.....	164
14.3.1	Start the CoDeSys Programming System.....	164
14.3.2	Creating a CoDeSys Project.....	165
14.3.3	Adding Library Management.....	166
14.3.4	Adding Libraries.....	168
14.3.5	Creating a Control Program.....	169
14.3.6	Creating a Task Configuration.....	171
14.3.6.1	Creating a Task.....	173
14.3.6.2	Assignment between POU and Task.....	175
14.3.7	Selecting the Communication Setting.....	176
14.3.8	Loading a PLC Program.....	177
14.3.9	Adding I/O Modules.....	179
14.3.9.1	Assignment Between the Process Data of the I/O Modules and the Variables of the Control Program.....	181
14.3.10	Accessing the MODBUS Register.....	183
14.3.10.1	Assignment Between MODBUS Registers and the Variables of the Control Program.....	186
14.3.11	Accessing EtherNet/IP Fieldbus Variables.....	188
14.3.11.1	Assigning the Fieldbus Variables in the Control Program.....	193
14.3.11.2	Assigning Fieldbus Variables to the EtherNet/IP Objects.....	195
14.3.12	Creating a Boot Project and Executing it at Fieldbus Coupler Start	197
14.4	Notes on CoDeSys Functions.....	199
15	Diagnostic and Status Information.....	200
15.1	Ethernet Status Messages via LED Signals.....	201
15.2	Operational Messages of the Fieldbus Coupler via LED Signals.....	203
15.3	Error messages from the fieldbus coupler via LED signals.....	205
15.3.1	Progression of blink sequence.....	206
15.3.2	Example of an error message via blink code.....	207
15.3.3	Meaning of the Blink Codes and Procedures for Troubleshooting..	208
15.4	Readout of Blink Codes using WAGO DTM.....	219

16	Parameterization via FDT/DTM	220
16.1	Installing the FDT/DTM Components	221
16.2	Starting WAGOframe	223
16.3	Expansion of Device Catalog to include 767 Components.....	224
16.4	Setting Up Network Manually	225
16.4.1	Adding the Communication DTM.....	225
16.4.2	Selecting the Communications Interface for WAGOframe	227
16.4.3	Adding a Fieldbus Coupler.....	228
16.4.4	Adding the I/O Modules	229
16.5	Online and Offline Configuration	230
16.5.1	Offline Configuration	230
16.5.2	Online Configuration.....	232
16.6	The "Additional Functions" and "Scan" Selections	234
16.6.1	Changing the Bus Address	235
16.6.2	Assigning the I/O Owner.....	237
16.6.3	Diagnostic Setup.....	240
16.6.4	Service Page	241
16.6.5	User Administration	242
16.6.6	File System	243
16.6.7	Creating a Network.....	244
16.6.8	Life List	245
16.6.9	System Update.....	247
16.6.9.1	Adding the DTM System Update	248
16.6.9.2	Go online to 767 Nodes using Update DTM	249
16.6.9.3	Updating the 767 Components	250
16.7	Parameterization.....	255
16.7.1	General Parameters.....	257
16.7.2	Fieldbus-Specific Parameters	263
16.7.3	Parameters of Inputs/Diagnostic Overview	267
16.7.4	Global Settings	269
16.7.5	Parameters of Field Supply	269
17	Maintenance and Service	270
17.1	Updating the Firmware	270
17.2	Replacing the Fieldbus Coupler	270
17.2.1	Disconnecting the Cables	270
17.2.2	Removing the fieldbus coupler from your system	271
17.2.3	Removing the Fieldbus Coupler from the Carrier Rail	271
17.2.4	Removing the Fieldbus Coupler from the Profile Adapter	272
17.2.5	Connecting a New Fieldbus Coupler.....	272
17.3	Disposal.....	272
18	Appendix.....	273
18.1	WAGO-Specific Classes	273
18.1.1	Bus Coupler Configuration Class Object (64 _{hex})	275
18.1.2	Discrete Input Point Classes	278
18.1.2.1	Discrete Input Point Class (65 _{hex})	278
18.1.2.2	Discrete Input Point Extended 1 (69 _{hex})	279
18.1.2.3	Discrete Input Point Extended 2 (6D _{hex}).....	280
18.1.2.4	Discrete Input Point Extended 3 (71 _{hex}).....	281

18.1.3	Discrete Output Point Classes	282
18.1.3.1	Discrete Output Point Class (66 _{hex})	282
18.1.3.2	Discrete Output Point Extended 1 (6A _{hex})	283
18.1.3.3	Discrete Output Point Extended 2 (6E _{hex})	284
18.1.3.4	Discrete Output Point Extended 3 (72 _{hex})	285
18.1.4	Analog Input Point Classes	286
18.1.4.1	Analog Input Point Class (67 _{hex})	286
18.1.4.2	Analog Input Point Extended 1 (6B _{hex})	287
18.1.4.3	Analog Input Point Extended 2 (6F _{hex})	288
18.1.4.4	Analog Input Point Extended 3 (73 _{hex})	289
18.1.5	Analog Output Point Classes	290
18.1.5.1	Analog Output Point (68 _{hex})	290
18.1.5.2	Analog Output Point Extended 1 (6C _{hex})	291
18.1.5.3	Analog Output Point Extended 2 (70 _{hex})	292
18.1.5.4	Analog Output Point Extended 3 (74 _{hex})	293
18.1.6	IOM Input Point Class (83 _{hex})	294
18.1.7	IOM Output Point Class (84 _{hex})	295
18.1.8	Node Information Class (82 _{hex})	296
18.1.9	Input Fieldbus Variables Classes	297
18.1.9.1	Input Fieldbus Variable USINT (A0 _{hex})	297
18.1.9.2	Input Fieldbus Variable USINT Extended 1 (A1 _{hex})	298
18.1.9.3	Input Fieldbus Variable USINT Extended 2 (A2 _{hex})	299
18.1.9.4	Input Fieldbus Variable UINT (A6 _{hex})	300
18.1.9.5	Input Fieldbus Variable UINT Extended 1 (A7 _{hex})	301
18.1.9.6	Input Fieldbus Variable UDINT (AA _{hex})	302
18.1.9.7	Input Fieldbus Variable UDINT Offset UINT (AB _{hex})	303
18.1.10	Output Fieldbus Variables Classes	304
18.1.10.1	Output Fieldbus Variable USINT (A3 _{hex})	304
18.1.10.2	Output Fieldbus Variable USINT Extended 1 (A4 _{hex})	305
18.1.10.3	Output Fieldbus Variable USINT Extended 2 (A5 _{hex})	306
18.1.10.4	Output Fieldbus Variable UINT (A8 _{hex})	307
18.1.10.5	Output Fieldbus Variable UINT Extended 1 (A9 _{hex})	308
18.1.10.6	Output Fieldbus Variable UDINT (AC _{hex})	309
18.1.10.7	Output Fieldbus Variable UDINT Offset UINT (AD _{hex})	310
18.2	Available Libraries	311
18.3	Accessories	312
18.3.1	S-BUS cable, assembled on both ends	312
18.3.2	S-BUS Terminator and USB Cable	312
18.3.3	Supply Cable Assembled on Both Ends	313
18.3.4	Ethernet Cable Assembled on One End	313
18.3.5	Ethernet Cable Assembled on Both Ends	314
18.3.6	Ethernet Accessories	314
18.3.7	Carrier Rail Adapters and Profile Adapters	314
18.3.8	Protective Caps	315
List of Figures		316
List of Tables		319

1 Notes about these Operating Instructions

The fieldbus coupler shall only be installed and operated in conjunction with these operating instructions and the system description.



WARNING

Observe release notes!

Please note that, within the SPEEDWAY system, a function is provided **without restriction** only if all system's components have the same system-wide firmware release. Therefore, always observe the appropriate release notes on products used.

NOTICE

Supply layout!

In addition to these operating instructions, you will need the "WAGO SPEEDWAY 767, System Description and Information" manual, which can be downloaded at www.wago.com. There you will find information regarding supply layout, etc.



Note

The operating instructions is part of the product and must be kept during the entire service life of the fieldbus coupler. The instructions must be passed to any subsequent owner or user of the fieldbus coupler. In addition, it must be ensured that any supplement to the instructions is also included with the instructions.

1.1 Validity of these Operating Instructions

These operating instructions are only applicable to the positive-switching WAGO SPEEDWAY 767 Series programmable fieldbus coupler, item number 767-2301. In this manual, the programmable fieldbus coupler will be referred to hereafter as simply the fieldbus coupler.

1.2 Explanation of Symbols



DANGER

Warning of physical injury

Indicates a direct hazard with a high level of risk which leads to death or serious physical injury if not avoided.



DANGER

**Warning of physical injury from electric current**

Indicates a direct hazard with a high level of risk which leads to death or serious physical injury if not avoided.



WARNING

Warning of physical injury

Indicates a possible hazard with a moderate level of risk, which may lead to death or (serious) physical injury if not avoided.



CAUTION

Warning of physical injury

Indicates possible hazards with a low level of risk, which could lead to minor or moderate physical injuries if not avoided.

NOTICE

Warning of damage to equipment

Indicates a possible hazard that could lead to equipment damage if not avoided.



NOTICE

Warning of damage to equipment from electrostatic discharge

Indicates a possible hazard that could lead to equipment damage if not avoided.



Note

Please note

Indicates possible malfunction, which does not lead to equipment damage if it is not avoided.



Information

Reference to additional information

Indicates other sources of information which are not an integral part of this documentation, such as the Internet.

2 Safety

2.1 Use in Compliance with Underlying Provisions

The EtherNet/IP fieldbus coupler provides digital process data from digital and analog I/O modules. The data is gathered by the fieldbus coupler and made available to a higher-level controller for further processing.

The fieldbus coupler shall not be used to control safety-related functions; i.e., emergency-off devices shall not be operated with this fieldbus coupler.

The fieldbus coupler shall only be used as a unit or in combination with I/O modules from the WAGO SPEEDWAY 767 Series.

The fieldbus coupler was developed for applications requiring IP 67 (NEMA type 6, 6P) protection.

The fieldbus coupler is expandable by a maximum of 64 I/O modules from the WAGO SPEEDWAY 767 Series.

Applications other than those described in this manual are not permitted.

2.2 Personnel Qualification

All sequences implemented on the fieldbus coupler may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current standards and guidelines for the fieldbus coupler and automation environment.

All changes to the controller should always be carried out by qualified personnel with sufficient sufficient skills in PLC programming.

2.3 Basic Safety Information

This section contains a summary of the most important warnings, which are also repeated in the individual sections. They serve as a protection to your health and a protection from equipment damage on the 767 Series components (fieldbus coupler and the I/O modules connected to it). Read and adhere to the following safety precautions before using the fieldbus coupler.



DANGER

Electric voltage!

Operate the 767 Series components exclusively with 24VDC PELV (Protective Extra Low Voltage) or SELV (Safety Extra Low Voltage) voltage sources. Failure to comply may result in electric shock.



CAUTION

Hot connection sockets!

Even when taking into account derating, high surface temperatures on the metallic connection sockets and on the enclosure can arise during operation. If the 767 Series component has been in operation, allow it to cool off before moving it.

NOTICE

The highest current carrying capacity of the supply contacts is 4A!

Always observe the maximum current carrying capacity per supply line (U_{LS} , U_A) for each 767 Series component and the overall power consumption for all 767 Series component. Neither of these values shall exceed 4A since an increase in current causes the contacts to overheat and damages the 767 Series components. Information regarding the power demand of each 767 Series component can be found in the corresponding data sheet, which is available from www.wago.com.

NOTICE

Exposed connections!

If connections have not been closed with protective covers, liquid or dirt can penetrate the fieldbus coupler and ruin it. Close all unneeded connections with protective caps to ensure that IP 67 degree of protection is provided.

- Disconnect the power supply from the system on which you wish to mount the 767 Series component.
- Always keep the cover cap of the DIP switch closed.
- Observe the appropriate accident prevention regulations for your system during assembly, startup, maintenance and repairs.
- The operating instructions of the 767 Series components and the system description must be readily available at the workplace.

- Observe the exact positioning (coding) between plug and socket.
- The 767 Series component shall not come in contact with substances having seeping and insulating properties. Otherwise, additional measures shall be taken for the devices, such as installation of an enclosure that is resistant to the above-mentioned substance properties.
- Electronic components fulfilling the ESD requirements according to the IEC 61000-6-2 are integrated in the 767 Series component. Since, under unfavorable circumstances, higher voltages may also occur due to electrical charge in the field, discharge must be ensured before performing work on the 767 system.
- Observe the correct layout of the potential equalization.
- Keep all cables a sufficient distance away from electromagnetic sources of interference in order to maintain a high level of interference resistance of the 767 system against electromagnetic emissions. Use only shielded cables at the necessary locations, and always observe the appropriate standards for EMC-suitable installations.
- For the power supply and for the S-BUS, use only pre-assembled WAGO system cables, so the specified characteristics of the technical data can be achieved.
- Replace defective or damaged 767 Series component (e.g., deformed connections), else function disruptions can occur in the respective fieldbus stations or nodes.
- When laying any cables, make sure that you do not lay them within the shear range of movable machine parts.
- For each activity, observe the corresponding personnel qualification in Section 2.2.
- Observe the marking on the front and rear side of the 767 Series component.

2.4 Safety Equipment

All products of series 767 are designed according to the IP 67 safety class. This consists of, among other things, complete touch protection of electric voltages and currents – even when wet.

2.5 Technical Condition of the 767 Series components

If any change is made to the 767 Series components or software and firmware without the written approval of WAGO Kontakttechnik GmbH & Co. KG, all liability claims are nullified. Parameterization shall only be carried out within the described scope (Section "Parameterization").

2.6 Notes on Operation

When integrating the 767 Series components in your machine or system, all the currently applicable norms, regulations and guidelines shall be observed during all activities. The emergency stop equipment shall remain effective in all operating modes of the system and machine.

For protection from electromagnetic interferences

- Connect your system to protective earth (PE), and
- Ensure that the cable routing and the installation of the fieldbus cable, S-BUS cable, supply cable, and sensor cable are correct.

The following elements for 24V supply shall be present:

- Outer lightning protection on buildings
- Inner lightning protection of supply lines and signal lines
- Safe electrical separation of low voltage 24VDC through PELV (Protective Extra Low Voltage) or SELV (Safety Extra Low Voltage) voltage sources

3 Information on Ethernet

Ethernet is a technology that has proven remarkably valuable and established for data transmission in information technology and in office communication. Even in the area of private PCs, Ethernet has experienced a worldwide breakthrough in a very short amount of time.

This technology was developed in 1979 by the companies Xerox, INTEL and DEC as a specification for a local network (LAN). A standardization (IEEE 802.3) took place in 1983. Ethernet predominantly uses coaxial cable or twisted two-wire line as a transmission medium. The components can be found almost anywhere and are very cost-effective. Ethernet can be connected to networks which are often preexistent (LAN, Internet) without any difficulty, and data transmission is carried out with a transfer rate of 10 Mbit/s or 100 Mbit/s.

In addition to the IEEE 802.3 standardization, Ethernet was also supplied with a higher-level communication software, TCP/IP (Transmission Control Protocol/Internet Protocol), for communication between systems. The TCP/IP protocol stack offers high dependability in information transmission.

Application protocols enables the user to create master applications through standardized interfaces and to transmit process data via an Ethernet interface.

By using Ethernet as a fieldbus, a uniform data transmission is established between production and office. Connecting the Ethernet fieldbus node to the Internet makes it possible to retrieve industrial process data worldwide for all types of applications, including from several nodes simultaneously if necessary. This facilitates a cross-locational inspection, visualization, remote maintenance and control of processes.

3.1 Physical Transmission into Network

3.1.1 Transfer rates

For data transmission, the Ethernet standard supports numerous technologies that differ in various characteristic values such as transfer rate, medium, segment length and transmission mode.

10Base2	Uses a coaxial cable (5 mm, 50 Ohm) for a 10 Mbit/s baseband signal for distances up to 185 m in a physical bus topology (often referred to as Thin Ethernet, ThinNet or Cheapernet).
10Base-T	Uses a twisted pair cable of type 24 AWG UTP or S-UTP for a 10 Mbit/s baseband signal for distances up to 100 m in a physical star topology.
100BaseTX	Specifies the 100 Mbit/s transmission on two pair cables via cables implemented with components of category 5. The cable, RJ45 wall sockets, patch panels, etc. must be laid out in accordance with this category for a minimum transmission frequency of 100 MHz.
100BaseFX	Point-to-point Ethernet connection where fiber-optic cables are utilized for data transmission.

In addition to those listed, there are other transmission standards such as 100Base-T4 (fast Ethernet via twisted pair cable) or P802.11 (wireless LAN) for a wireless transmission.

3.1.2 Network Topology

With 10Base-T and 100Base-T, several nodes are wired in a point-to-point manner, according to the Ethernet standard. For this reason, only the star topology and, for larger networks, the structure of a tree topology should be examined more closely. A computer network is created using components such as hubs and switches.

Star topology

The star topology is a network in which all nodes are connected to a central node. For this purpose, a hub is connected to a bus architecture like a normal PC, and the bus runs inside the hub.

Aside from the simple implementation, the advantages of such an application lie in the expandability of an existing network. A node can be added or removed without the risk of network failure. In addition, only the communication to the respective node is affected in the case of a defective line, thus increasing considerably the failure safety of the entire network.

With the star topology, administratively matching groups can very easily be formed, combined in hierarchical levels and linked in a tree-like structure.

Tree topology

The tree topology is a structure which is used for larger networks, companies or buildings. In this topology, various smaller networks are connected to each other hierarchically like a tree (branches, limbs and trunk), e.g. via routers.

Guidelines for cabling

General guidelines for setting up a LAN network are provided in "Structured Cabling". There you will find the maximum cable lengths allowed for cabling the grounds, building and levels.

The "Structured Cabling" is standardized in the standards EN 50173, ISO 11801 and TIA 568-A. It forms the basis for a future-oriented, cost-effective network infrastructure independent of application.

Additional information on industrial cabling can be found at www.iaona.de and www.odva.org.

3.1.3 Transmission modes

Ethernet supports transfer rates of 10 Mbit/s or 100 Mbit/s in full-duplex and half-duplex mode. To ensure a safe and fast transmission, the fieldbus coupler and its communication partner (link partner) must be configured for the same transmission mode.

A faulty configuration of the transmission mode can result in a loss of the communication partner, poor network performance or faulty performance of the fieldbus coupler.

The IEEE 802.3u ETHERNET standard offers two possibilities for configuring the transmission modes:

- Static configuration
- Dynamic configuration

Static configuration of transmission mode

Static configuration allows both communication partners to be set to one static transfer rate and duplex mode. The following configurations are possible:

- 10 Mbit/s, half-duplex
- 10 Mbit/s, full-duplex
- 100 Mbit/s, half-duplex
- 100 Mbit/s, full-duplex

Dynamic configuration of transmission mode

The second configuration option is the autonegotiation mode which is defined in the IEEE 802.3u standard. Using this mode, the transfer rate and the duplex mode are negotiated dynamically between both communication partners.

Autonegotiation allows the device to automatically select the optimum transmission mode.

To ensure a correct dynamic configuration process, the operation mode for the autonegotiation of both communication partners must be supported and activated. In some applications, it is beneficial to use the static configuration.

Errors during configuration of transmission mode

The following is a list of invalid configurations:

Problem	Cause	Symptoms
Mismatch of the transfer rate	Occurs if one link partner has been configured with 10 Mbit/s and the other with 100 Mbit/s.	Link failure
Duplex mode mismatch	Occurs when one link partner is running in full-duplex and the other in half-duplex mode.	Faulty or discarded data packets as well as collisions on the medium.
Mismatch during autonegotiation	Occurs when one link partner is running in auto-negotiation mode and the other is using a static configuration of the transmission mode in full-duplex operation.	The link partner, which is in autonegotiation mode, determines the network speed via the parallel detection procedure and sets the duplex mode to half-duplex. If the device is operating in full-duplex mode with static configuration, a duplex mode mismatch will occur (see above).

3.1.4 Ethernet hardware address (MAC-ID)

During production, each Ethernet device is equipped with a single, globally unique, physical Ethernet address, also called MAC-ID (Media Access Control Identity). This address can be used by the network operating system for addressing on the hardware level.

The address has a fixed length of 6 bytes (48 bits) and contains the address type, manufacturer's ID and the serial number.

Example of MAC-ID: 00_H.30_H.DE_H.00_H.00_H.01_H.

The addressing of separate networks is not possible with Ethernet. Therefore, higher-level protocols must be used if an Ethernet network is to be connected with other networks.

If two or more data networks are to be connected with each other, routers must be utilized.

Ethernet data packet

The data telegrams exchanged on the transmission medium are referred to as "packets" or "Ethernet packets". The transmission occurs connectionless, i.e. the transmitter receives no confirmation from the receiver. The user data is packed into a frame of address information. The structure of one such packet is shown in the following figure.

Preamble	Ethernet header	Ethernet user data range	Checksum
8 bytes	14 bytes	46-1500 Byte	4 bytes

The preamble synchronizes between the transmitting station and receiver station. The Ethernet header contains the MAC addresses of the transmitter and receiver and a type field. The type field identifies the subsequent protocol using a unique coding (e.g. 0800_{hex} = Internet Protocol).

3.2 Logical Addressing

3.2.1 IP – Internet Protocol

The Internet Protocol separates data telegrams into segments and is responsible for transporting them from one network station to the other. During this process, the participating nodes can either be located in the same network or in different physical networks that are connected to each other with routers.

The routers are able to select various network transmission paths through a network connection, thus avoiding overloads and disruptions of individual networks.

However, this includes the possibility of certain segments being chosen that are shorter than others, thus allowing the data telegrams to overtake each other and falsifying the sequence of the data packets.

For this reason, guaranteeing a correct transmission must take place at higher levels, e.g. through TCP.

IP data packets

The IP data packets include an abundance of address and additional information in the "Packet Header" aside from the user data which is to be transported.

IP header	IP user data range

The most important pieces of information in the IP header are the IP addresses of the transmitter and receiver, as well as the transport protocol used.

3.2.2 IP addresses

For communication in the network to take place, each fieldbus node must possess a 32-bit Internet address (IP address). Internet addresses must be unique across the entire network connection.

As shown below, there are various address classes with a network identification (net ID) in differing lengths and a host computer identification (host ID). The net ID defines the network in which the user is located. The host ID identifies a particular user within this network.

For purposes of addressing, the networks are divided into several network classes:

Class A: (Net ID: Byte1, Host ID: Byte2 - Byte4)

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'.
 In other words, the highest byte can lie in the range between '0 0000000' and '0 1111111'.

The address range of Class A networks therefore always lies between 0 and 127 in the first byte.

Class B: (Net ID: Byte1 - Byte2, Host ID: Byte3 - Byte4)

e.g.: 181 . 16 . 232 . 22

10110101	00010000	11101000	00010110
10	Net ID	Host ID	

↑ The highest bits in the Class B networks are always '10'.

In other words, the highest byte can lie in the range between '10 000000' and '10 111111'.

The address range of Class B networks therefore always lies between 128 and 191 in the first byte.

Class C: (Net ID: Byte1 - Byte3, Host ID: Byte4)

e.g.: 201 . 16 . 232 . 22

11000101	00010000	11101000	00010110
110	Net ID		Host ID

↑ The highest bits in the Class C networks are always '110'.
In other words, the highest byte can lie in the range between '110 00000' and '110 11111'.

The address range of Class C networks therefore always lies between 192 and 223 in the first byte.

Other network classes (D, E) are used for special functions.

Key data for the IP addresses

	Address range of network component	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})	Approx. 65 thousand (2^{16})
Class C	192.000.000.XXX - 223.255.255.XXX	Approx. 2 million (2^{21})	254 (2^8)



Note

Never set all bits in a byte to 0 or 1 (byte = 0 or 255). These numbers are reserved for special functions and may not be assigned. For example, 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, universally unique IP addresses are used which have been assigned by a central domain name registrar. In Germany, for example, the assigning takes place via the DE NIC (Deutsches Netzwerk Informations Center) in Karlsruhe.

3.2.3 Subnet

To enable routing within large networks, a convention was established in the RFC 950 specification in which a part of the Internet address, the host ID, is subdivided into a subnet number and the actual station number of the node. Using the network number, branching into internal subnetworks can now occur within the subnetwork. However, the entire network is externally visible as a unit. Although the size and status of the subnetwork ID are not specified, the size is dependent on the quantity of subnets to be addressed and the quantity of hosts per subnet.

1	8	16	24	32
1	0	Net ID	Subnet ID	Host ID

Subnet mask

The so-called subnet mask was established for coding the subnets in the Internet. The subnet mask is a "bit mask" with which special bits of the IP address can be hid or selected. The mask defines which bits of the host ID are used for the subnet coding and which constitute the ID of the host. The entire IP address range theoretically lies between 0.0.0.0 and 255.255.255.255.

The 0 and the 255 are each reserved from the IP address range for the subnet mask.

The standard masks which are dependent on the respective network class appear as follows:

- **Class A subnet mask:**

255	.0	.0	.0
- **Class B subnet mask:**

255	.255	.0	.0
- **Class C subnet mask:**

255	.255	.255	.0

Depending on the division of the subnet, the subnet masks can also include other values in addition to 0 and 255, such as 255.255.255.128 or 255.255.255.248, etc. The subnet mask number is assigned to you by your network administrator. In conjunction with the IP address, this number determines to which network your PC and your Ethernet node belong.

The transmitter node located in a subnet first computes the correct network address from its own IP address and from the subnet mask.

The transmitter node only checks the node number when it is connected; if the number is correct, it then reads out the entire packet border.

The following is an example of 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

The network mask specified by the administrator must be indicated just like the IP address when installing the network protocol.

3.2.4 Gateway

The subnets of the Internet are typically connected via gateways. These gateways forward IP packets to other networks or subnets.

For a PC or fieldbus node connected to the Internet, this means that the correct IP address of the standard gateway must be indicated, in addition to the IP address and network mask for each network card. This IP address should likewise be provided by your network administrator.

Without an indication of this address, the IP functionality remains limited to the local subnet.

3.2.5 RAW IP

RAW IP does without protocols such as PPP (point-to-point protocol). In RAW IP, the TCP/IP packets are exchanged directly without handshaking, making possible a faster connection setup. However, configuration with a fixed IP address must have already taken place. Advantages of RAW IP are a high data transfer rate and good stability.

3.2.6 IP Multicast

Multicast refers to a transmission mode from one point to a group, called a point-to-multi-point transmission or a multi-point connection. The advantage of Multicast is that messages can be transmitted simultaneously to several users or closed user groups using one address.

IP Multicasting on the internetworking level is carried out through the "Internet Group Message Protocol" (IGMP). This protocol is used by neighboring routers to mutually inform the other about group assignments.

When distributing Multicast packets in the subnet, the IP assumes that the Data Link Layer provides access to Multicasting. In the case of Ethernet, Multicast addresses exist with which a packet addressed by them is sent to several recipients via a single sending operation. This rests on the fact that a common medium has the possibility of sending packets to several recipients *simultaneously*. The nodes do not need to inform each other as to what belongs to a Multicast address – each node receives each packet physically.

The address dissolution of IP address to Ethernet address is triggered algorithmically; IP Multicast address are embedded in Ethernet Multicast addresses.

3.3 Transport protocols

TCP protocol

Set up on the Internet protocol, the TCP (Transmission Control Protocol) assumes the security of the data transport through the network.

For this purpose, the TCP establishes a connection between two users for the duration of the data transmission. The communication is carried out in full-duplex mode: both users can send and receive data simultaneously.

The transmitted user data is provided with a 16-bit checksum by the TCP, and each packet contains a sequence number.

The recipient checks whether the packet has been correctly received using the checksum and subsequently clears the sequence number. This results in an acknowledgement number which is sent back with the next self-sent packet as acknowledgement.

This guarantees that the loss of TCP packets is detected and that these packets can be resent in the correct sequence, if necessary.

TCP port numbers

The TCP can also target a specific application (service) on the addressed host supplemental to the IP address (network and host address). For this purpose, the applications in a host (such as the web server, FTP server and others) are addressed via differing port numbers. For known applications, fixed ports are assigned which correspond to each application when setting up the connection.

Examples:

Telnet	Port number: 23
HTTP	Port number: 80

A comprehensive list of the "standard services" can be found in the RFC 1700 specifications (1994).

TCP data packet

The packet header of a TCP data packet consists of at least 20 bytes and includes the port number of both the sender's and recipient's application, the sequence number, the acknowledgement number, etc.

The TCP packet thus originating is implemented into the user data range of an IP packet so that a TCP/IP packet is generated.

TCP header	TCP user data range

UDP

Like the TCP protocol, the UDP protocol is responsible for data transmission. In contrast to the TCP protocol, UDP is not connection-oriented; that is, there are no control mechanisms for the data exchange between sender and receiver. The advantage of this protocol lies in the efficiency of the transmitted data, and therefore in the resultant higher processing speed.

ARP

ARP (short for "Address Resolution Protocol").

This protocol connects the IP address with the physical MAC address of the respective Ethernet card. It is always used when the data transmission to an IP address takes place within the same logical network as the sender.

3.4 Configuration and Diagnostic Protocols

BootP

The BootP protocol defines a mechanism which is used to assign a fixed IP address via the MAC ID of a device.

The fieldbus coupler is enabled to send requests into the network and to retrieve the necessary network information, such as the IP address of a BootP server.

The BootP server waits for incoming BootP requests and generates the corresponding reply from a configuration database.

Dynamic configuration of the IP address via a BootP server provides the user with a flexible and easy way to configure the network.

The BootP Client assists in configuring the network parameters:

Parameter	Meaning
IP address of the Client	Network address of the fieldbus coupler
IP address of the router	If communication takes place outside the local network, the IP address of the router (gateway) is given in this parameter.
Subnet Mask	The subnet mask makes it possible for the fieldbus coupler to distinguish which parts of the IP address identify the network and which identify the network stations.
IP addresses of the DNS server	The IP addresses from a maximum of 2 DNS servers can be specified here.
Host name	Name of host
Time offset	For the time zone
Domain name	Name of device
NTP server	Network time protocol server (SNTP)

DHCP

The "Dynamic Host Configuration Protocol" defines a question/answer mechanism for network users with which the MAC ID of a fieldbus node can be assigned a temporary IP address. For this purpose, a network node is enabled to send requests into the network and to retrieve the necessary network information, such as the IP address of a DHCP server.

The DHCP server waits for incoming DHCP requests and generates the reply from a configuration database.

Dynamic configuration of the IP address via a DHCP server provides the user with a flexible and easy way to configure the network.

The DHCP Client assists in the dynamic network configuration of the fieldbus coupler by setting the following parameters:

Parameter	Meaning
IP address of the Client	Network address of the fieldbus coupler
IP address of the router	If communication takes place outside the local network, the IP address of the router (gateway) is given in this parameter.
Subnet Mask	The subnet mask makes it possible for the fieldbus coupler to distinguish which parts of the IP address identify the network and which identify the network stations.
IP addresses of the DNS server	The IP addresses from a maximum of 2 DNS servers can be specified here.
Lease time	The maximum duration, i.e. how long the fieldbus coupler maintains the assigned IP address, can be defined here.
Renewing time	The renewing time indicates when the fieldbus coupler must renew the lease time.
Rebinding time	The rebinding time indicates after what amount of time the fieldbus coupler must have received its new address.
Time offset	For the time zone
Domain name	Name of device
NTP server	Network time protocol server (SNTP)

For configuration of the network parameters via the DHCP, the fieldbus coupler sends an inquiry independently to the DHCP server after initialization. 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 released via the I/O LED.

When using a lease time, the values for the renewing time and rebinding time are also indicated. After the renewing time has elapsed, the fieldbus coupler attempts to automatically renew the lease time for its IP address.

If this is unsuccessful when the rebinding time has elapsed, the fieldbus coupler attempts to receive a new IP address. The renewing time should be approximately half of the lease time. The rebinding time should be approximately $\frac{7}{8}$ of the lease time.

DNS

By way of the DNS server, the DNS Client makes it possible to convert logical Internet names, such as www.wago.com, into the corresponding decimal IP address displayed with cut-off points. The addresses of the DNS server are configured using DHCP or the web-based management. Up to two DNS servers can be specified. The host identification can take place with two functions; an internal host table is not supported.

SNTP

The SNTP Client is used for synchronizing the time between a time server (NTP and SNTP servers, versions 3 and 4) and the time module integrated in the fieldbus coupler. The protocol is executed via a UDP port. Only the Unicast addressing is supported.

The SNTP Client is configured via the web-based management. The following parameters must be set:

Parameter	Meaning
Address of the time server	The address assignment can either be done via an IP address or a host name.
Time zone	To operate the fieldbus coupler with SNTP in various countries, you must specify a time zone. Setting the time zone is relative to GMT (Greenwich Mean Time). A range from -43200 to +50400 seconds can be specified (-12 hrs to +14 hrs).
Update time	The update time specifies the interval in seconds in which the synchronization with the time server should occur.
Enable time client	Specifies whether the SNTP client should be activated or deactivated

HTTP

The "Hypertext Transfer Protocol" is used by WWW (World Wide Web) to transfer hypermedia, text, pictures, audio files, etc.

HTTP currently forms the basis of the Internet and, like the BootP protocol, is based on requests and replies.

The HTTP server implemented on the fieldbus coupler reads out the saved HTML pages. The HTML pages can give information about the fieldbus coupler (status, configuration) or the network.

On some HTML pages, fieldbus coupler settings can also be set or changed via the web-based management, e.g. whether the network configuration of the fieldbus coupler should take place via the DHCP, the BootP protocol or from the stored data in EEPROM. The HTTP server uses the port number 80.

FTP

The file transfer protocol makes it possible to exchange files between various network users regardless of the structure of the operating system.

The FTP saves the HTML pages created by the user, the IEC 61131 program and the IEC 61131 source code to the fieldbus coupler and reads them out.

4 Information on EtherNet/IP

EtherNet/IP stands for "Ethernet Industrial Protocol" and defines an open industry standard which expands the classic Ethernet to include an industry protocol. This standard was developed collectively by "ControlNet International" (CI) and the "Open DeviceNet Vendor Association", (ODVA) with support from the "Industrial Ethernet Association" (IEA).

This communication system makes it possible for devices to exchange time-critical application data in an industrial environment. The device spectrum ranges from simple I/O devices (e.g. sensors) to complex controls (e.g. robots).

EtherNet/IP is based on the TCP/IP protocol family. Therefore, it takes over the lower four layers of the OSI layer model in unmodified form so that all standard Ethernet communication modules (PC interface cards, cables, connectors, hubs, switches, etc.) can likewise be used with EtherNet/IP.

Above the transport layer, the "Encapsulation Protocol" can be found with which the "Common Industrial Protocol" (CIP) is used on the TCP/IP and UDP/IP. CIP is already being used as a large, network-independent standard for ControlNet and DeviceNet. Transferring an application on one of these systems can therefore be easily achieved. The data exchange is based on an object model. Thus, ControlNet, DeviceNet and EtherNet/IP all have the same application protocol and can therefore collectively use device profiles and object libraries. These objects make possible a plug-and-play interoperability between complex devices from different manufacturers.

OSI model of the CIP protocol

To clarify the commonalities between DeviceNet, ControlNet and EtherNet/IP, the following representation shows the classification of the ISO/OSI communication model.

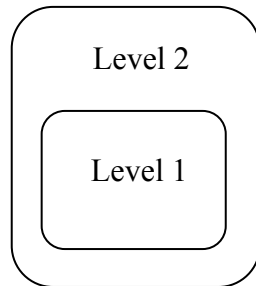
7 Application layer	Object library (Communications, applications, time synchronization)			Safety object library	Common Industrial Protocol (CIP)
6 Presentation layer	Data management services Explicit and I/O messages			Safety services and messages	
5 Session layer	Connection management, routing				
4 Transport layer	TCP/UDP	CompoNet Network in transport	ControlNet Network in transport	DeviceNet Network in 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	

Properties 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.

Level 1: explicit messages – server

Level 2: level 1 + I/O messages – server



- UCMM compatible (connectionless, client and server)
- 128 "Encapsulation Protocol Sessions"
- 128 class 3 connections or class 1 (combined)

Class 3 connection – explicit messages
(connection-oriented, client und server)

Class 1 connection – I/O messages
(connection-oriented, client und server)

5 Information on MODBUS

The MODBUS protocol was developed in 1979 and today, MODBUS TCP is an open Internet Draft Standard of the IETF (Internet Engineering Task Force).

The tried and tested original MODBUS services as well as the object model have not been modified and have been adopted by TCP/IP as a transmission medium. Communication takes place via port 502, which is reserved for MODBUS. The MODBUS family thus consists of the classic MODBUS RTU and MODBUS ASCII (asynchronous transmission via RS 232 or RS 485) and MODBUS TCP (connection-oriented client-server communication via Ethernet).

With the MODBUS protocol, any memory location in the process image of the fieldbus coupler can be read out or modified via a higher-level controller or another user. However, the node structure determines which I/O module is located at a specific memory location.

WAGO is expanding the MODBUS family to include MODBUS UDP. This version uses a connectionless, asynchronous client server communication via Ethernet.

MODBUS works with the following basic data types:

- Discrete (digital) inputs,
- Coils (digital outputs),
- Input register (16-Bit input data) and
- Holding register (16-Bit output data).

MODBUS differentiates between digital services and register services. The digital MODBUS services are also called Coil services.

The digital MODBUS services can only determine or change the state of digital I/O modules. Analog I/O modules are ignored or cannot be reached. A data exchange with analog I/O modules is carried out using register services.

The specification identifies the "big endian" format for the byte order in MODBUS telegrams. In this format, the most significant byte (MSB) is written to the lowest memory address. If the telegrams are analyzed on a PC, the byte order must be converted to "little endian" format.

6 Fieldbus Coupler Description

This section contains a short overview of the fieldbus coupler structure and the most important features. Detailed information can be found in the appropriate sections.

The fieldbus coupler assists in attaching the 767 series to the Ethernet. The fieldbus coupler can be used to program programs according to IEC 61131-3, through which it is independently operable. This means that the fieldbus coupler can be operated both with and without a higher-level controller. The fieldbus coupler is based on a 32-bit CPU with multitasking capabilities; i.e., several programs can be executed simultaneously.

A transfer rate of 10 Mbit/s or 100 Mbit/s is possible with the fieldbus coupler in both half-duplex and full-duplex operation. Two fieldbus coupler ports are connected internally via an Ethernet switch. Therefore, the Ethernet fieldbus coupler can be integrated not only in star topology, but also in line topology. The maximum distance between two fieldbus couplers is 100 m. Up to 20 fieldbus couplers are cascable, such that a maximum line of 2 km can be achieved.

An internal server is available for Web-based applications. In essence, HTML pages stored in the fieldbus coupler allow access to information on the configuration and status of the fieldbus node via Web browsers. It is also possible to store individual HTML pages using the implemented file system or to start programs directly.

Application programs can be created with the CoDeSys 3 programming system according to IEC 61131-3. For the IEC 61131-3 programming, the fieldbus coupler provides 1024 KB of program memory, 256 KB of data memory and 32 KB of retained memory. You can use function blocks to program the clients and the server via an internal socket API for all transport protocols (TCP, UDP, etc.). Library functions are available for function expansion.

The EtherNet/IP and MODBUS TCP/UDP application protocols are provided to you to exchange process data. You can use these application protocols either individually or combined. Moreover, the BootP, DHCP, HTTP, SNTP and FTP protocols are implemented in the fieldbus coupler. These protocols can be configured via the web-based management and using the "WAGOframe" FDT/DTM frame application. Please see Sections 10.2 and 16 for more information.

6.1 Connections

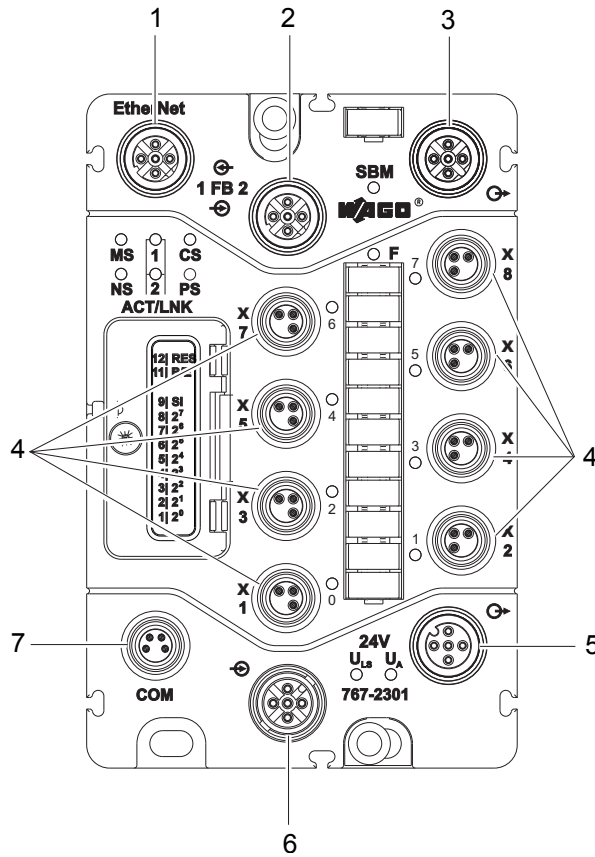


Figure 1: Identification of connections

Position	Description	Function
1, 2	Ethernet connections, M12 plugs, D-coded	Physical connections for the integration of the fieldbus coupler into the Ethernet network.
3	S-BUS output M12 socket, B-coded	Physical connection for connecting I/O modules to the S-BUS and for closing the S-BUS.
4	Digital inputs 1 – 8 M8 socket	For connecting digital sensors (e.g., initiators or limit switches).
5	Supply output M12 socket, A-coded	Use of the system and/or field supply for the following I/O module.
6	Supply input M12 plug, A-coded	Infeed of system and field power supply
7	USB interface M8 socket, 4 poles, A-coded	Parameterization independent of fieldbus, configuring and diagnosing entire fieldbus station and updating device software.

6.2 Marking Possibilities and Fasteners

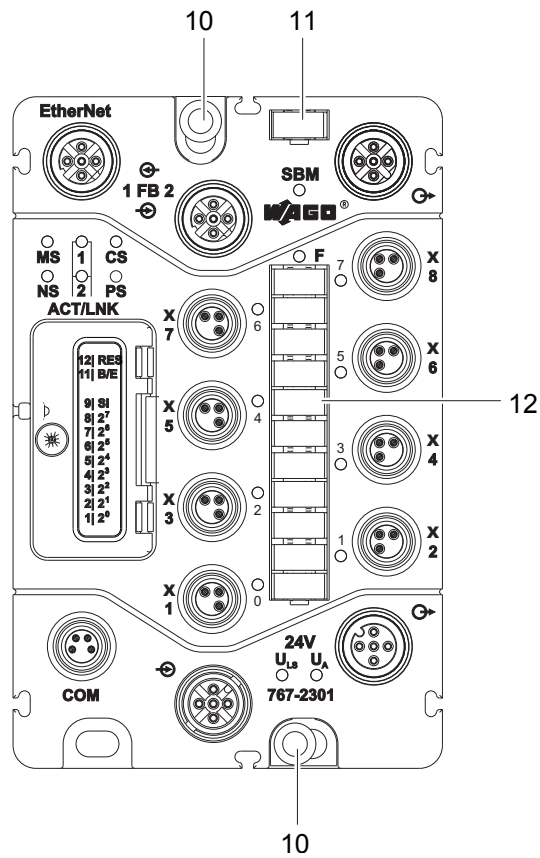


Figure 2: Identification of possibilities for marking and fastening

Position	Description	Function
10	Mounting holes for M4 screws	For fastening and grounding the fieldbus coupler
11	Module marking plate	For identification of the fieldbus coupler within the Ethernet network.
12	Marker strips	For identifying inputs

6.3 LEDs and Operating Elements

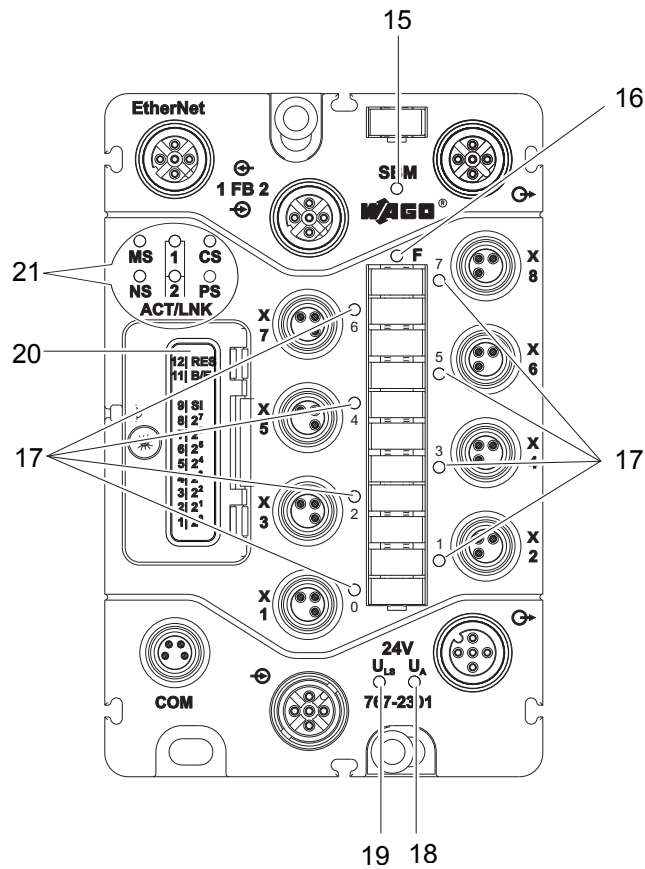


Figure 3: Identification of LEDs

Position	LED/ Operating element	Color	Meaning
15	SBM	Green, red	Status of S-BUS Master
16	F	Red	Diagnostic information of digital inputs is available.
17	LEDs 0 to 7	Yellow	Input signal pending
18	U_A	Green	Actuator supply is present
19	U_{LS}	Green	Logic supply and sensor supply are present.
20	Address and operating mode switch (DIP switch)	-	Setting the IP address. Executing a hardware reset.

Position	LED/ Operating element	Color	Meaning
21	MS	Green, red	Module status
	NS	Green, red	Network status
	ACT/LNK 1	Green	Status of the physical connection to the Ethernet network (FB 1)
	ACT/LNK 2	Green	Status of the physical connection to the Ethernet network (FB 2)
	CS	Green, red	Fieldbus coupler status
	PS	Green, red	Program status of run-time system

Detailed information regarding the LEDs can be found starting in Section 15.

6.4 Address and Operating Mode Switch (DIP Switch)

The IP address of the fieldbus coupler can be set with the DIP switch. Moreover, the DIP switch makes it possible to execute a hardware reset of the fieldbus coupler. A transparent cover protects the DIP switch. The assignment of values is as follows:

Table 1: Default settings of DIP switch

Switch	1	2	3	4	5	6	7	8	9	10	11	12
Binary value/ Functions	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	Set Network	Run/ Stop	Boot/ Execute	Reset
Switch setting	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off

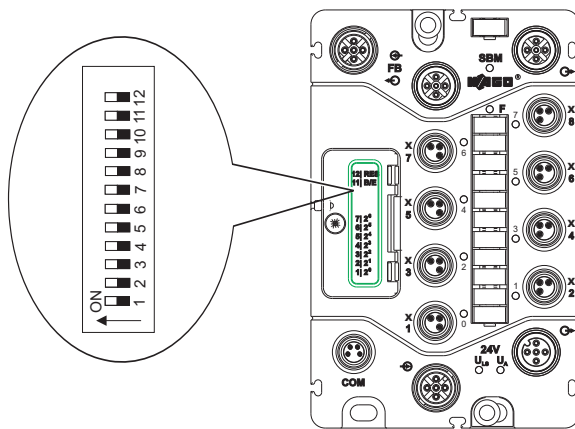


Figure 4: DIP switch. All switches are set on "Off" at delivery.

Table 2: Explanation of DIP switch

Switch	On	Off
1 ... 8	Set the IP address (least significant byte, LSB) of the fieldbus coupler using this switch. Addresses in the range of 1 to 254 are permitted. Addresses 0 and 255 are reserved and may not be used. This functionality is controled via switch 9 (on/off).	
9	The IP address (LSB) is defined via switches 1 ... 8.	The assignment of IP addresses is based on the mode (DHCP, BootP, ...) which you configured and selected in the web-based management or WAGOframe.
10	This switch is used to specify whether the boot project is to be automatically executed when the fieldbus coupler is restarted.	
11	This DIP switch must always be set to „Off“.	
12	This switch is used to initiate a hardware reset of the fieldbus coupler. You can override this status by returning the switch to "Off."	Normal operation.

Detailed information can be found in Section 9.4.

6.5 Labeling and Symbols on the Back Side

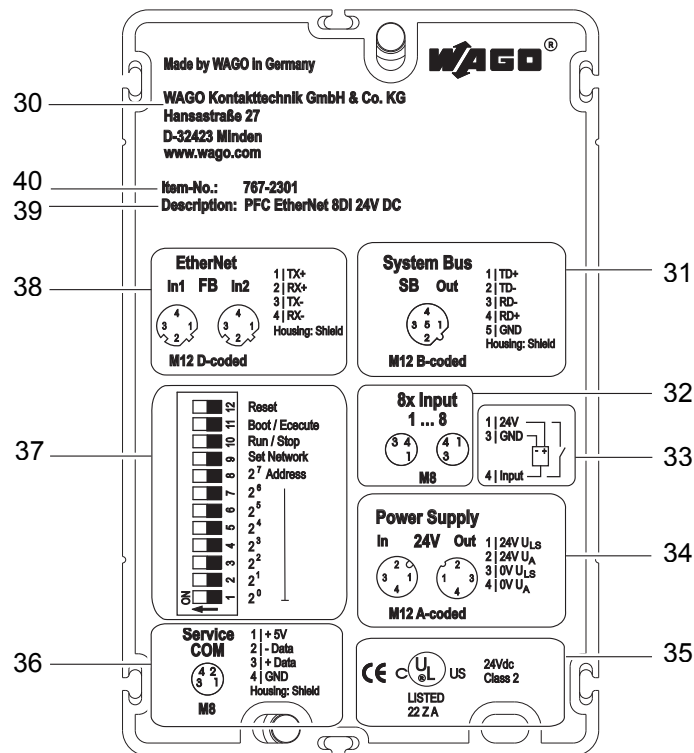


Figure 5: Marking and symbols

Position	Description
30	Manufacturer's mailing address
31	Connection assignment of S-BUS
32	Connection example
33	Connection assignment of digital inputs
34	Connection assignment of supply input and output
35	Information on approvals and CE marks
36	Connection assignment of USB interface
37	Labeling and assignment of DIP switch
38	Connection assignment of Ethernet connections
39	Clear labeling of fieldbus coupler
40	Order number

6.6 Lateral Marking

On the side of the fieldbus coupler is a label, which gives information that would prove useful in the case of a complaint:

- BA: Work order number (50)
- SN: Serial number (50)
- MAC address (51)
The MAC address assists in identifying and addressing Ethernet devices. Each MAC address is unique worldwide.
- Manufacturing number (52)

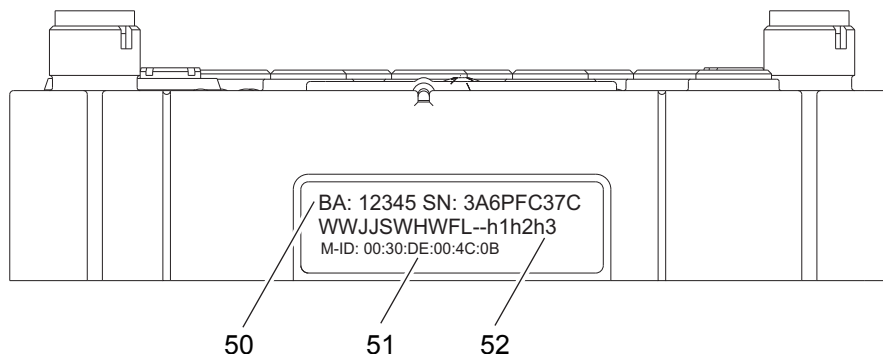


Figure 6: Label on fieldbus coupler

Table 3: Description of manufacturing number

Abbreviation	Description
WW	Week of production
JJ	Year of production
FW	Firmware release index When updating the firmware, please note that the firmware release index may not be conformed to the printed firmware release index on the side of the fieldbus coupler. The “Electronic Type Label” (section 9.1) shows the actual fimware release index.
HW	Hardware release index
FL	Firmware loader release index
h1h2h3	Internal manufacturer information

6.7 Block diagram

The following schematic representation provides an overview of the power supply and principle of operation of the power supply connections, as well as the digital inputs of the fieldbus coupler (see also chapters "Connecting the Supply Cable" and "Connecting the Sensor Cable").

Please note that the common power supply of the sensors is distributed to all connections (X1 – X8, Pin 1) of the fieldbus coupler.

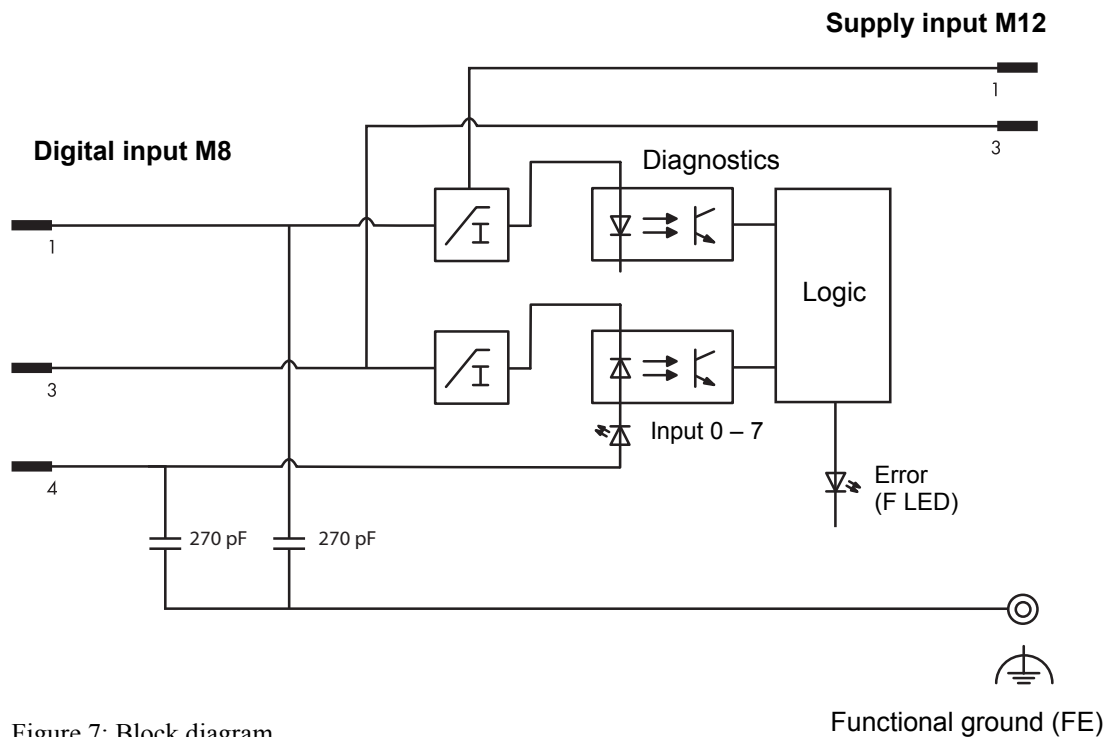


Figure 7: Block diagram

6.8 Dimensions

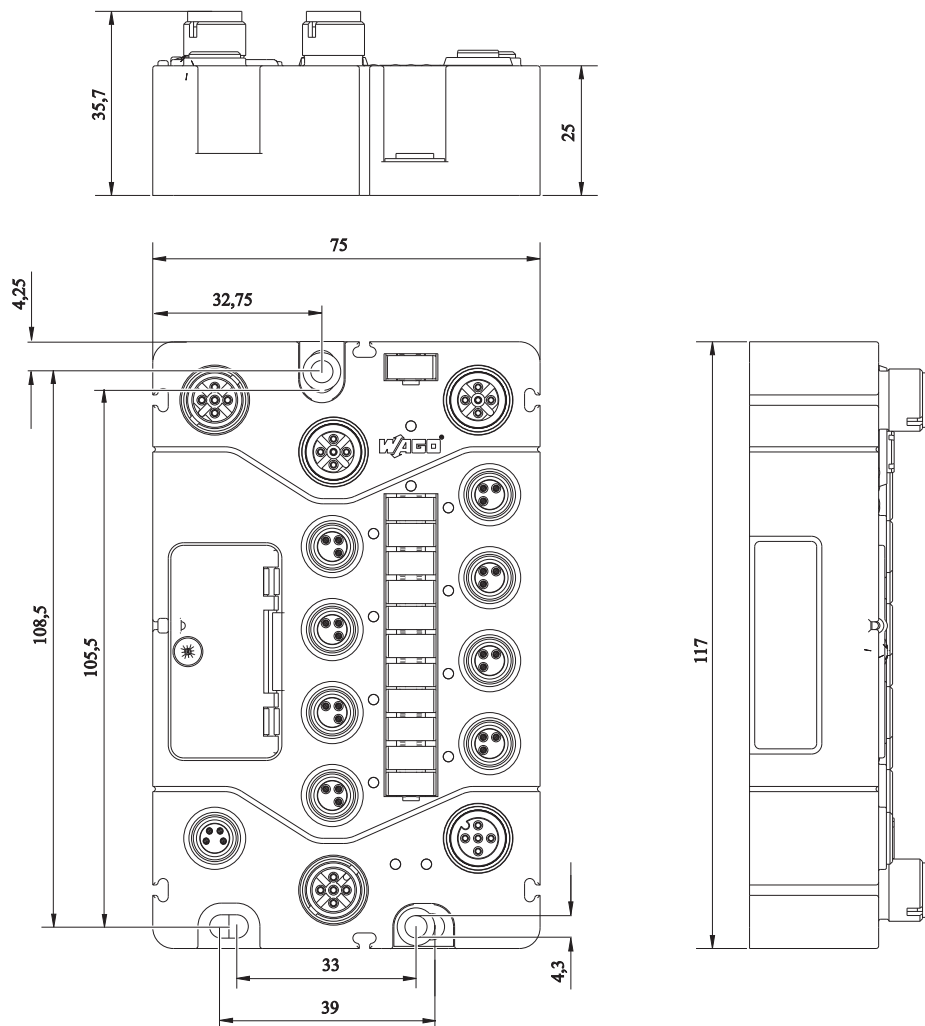


Figure 8: Dimensions of fieldbus coupler in mm

6.9 Technical Data

Table 4: Technical data

Fieldbus	
Device type	ETHERNET device
Connection type	M12 connectors, D coded, 5 poles
Baud rate	10/100 Mbit/s
Transmission medium	Copper cable
Station address	1 – 255 (last byte of IP address adjustable via operation panel)
Protocols	MODBUS/TCP (UDP), ETHERNET/IP
Programming	
CoDeSys 3	Development system for programming and visualization according to IEC61131-3
Module supply	
Connection type	M12 connectors, A coded, 4 poles*
Current carrying capacity of the supply connections	Maximum 8 A (V_{LS} : 4 A, V_A : 4 A)
Supply voltage	DC 24 V (-25 % ... +30 %)
Logic and sensor voltage U_{LS}	
Actuator voltage U_A **	DC 24 V (-25 % ... +30 %)
Supply current	Typically 125 mA + sensors (max. 400 mA)
Logic and sensor current I_{LS}	
Actuator current I_A	5 mA
Protection	Reverse voltage protection for $V_{LS} + V_A$ Short-circuit protection for sensor supply
Digital inputs	
Number of inputs	8
Connection type	M8 connectors, 3 poles
Connection system	2- to 3-wire
Input filter	Parametrizable
Input characteristic	Type 1, acc. to IEC 61131-2
Signal voltage 0	DC -30 V ... +5 V
Signal voltage 1	DC +15 V ... +30 V
Input wiring	High-side switching
Input voltage	24 V DC (-30 V DC < V_{IN} < +30 V DC)
Input current	Typically 2.8 mA
Cable length, unshielded	≤ 30 m
Wrong connection of inputs	No effect
S-BUS	
Number of expendable modules	64
Connection type	Shielded M12 connector, B coded, 5 poles

* Derating should be observed

** Also required for supply transmission

Table 4: Technical data

Isolation	
Channel - Channel	No
U_{LS} , U_A , S-BUS, fieldbus	500 V DC each
Service	
Type	USB
Connection type	M8 connectors, 4 poles
Standards and approvals	
UL 508	
Conformity marking	CE
Configurable functions	
Fieldbus coupler	See section parameterization
Digital inputs	
Input filter (per channel)	0.1/ 0.5/ 3 /15 /20ms/ filter off
Inversion (per channel)	on/off
Online simulation (per channel)	Lock/unlock; simulation value: 0/1
(per module)	diagnostics
I/O diagnostics	
Per module	Short circuit/wire break of sensor supply
Per module	Undervoltage ($U_{LS} + U_A$)
Process image	
Input process image	2048 bytes
Output process image	2048 bytes
Input variables	512 bytes
Output variables	512 bytes
Program memory	1024 KB
Data memory	256 KB
Retained memory	32 KB (20 KB retain, 12 KB flag)
Indicators	
MS: EtherNet module status	LED (green/red)
ACT/LNK 1 : Ethernet Data exchange/network connection	LED (green)
CS : Controller status	LED (green/red)
NS : Ethernet network status	LED (green/red)
ACT/LNK 2 : Ethernet Data exchange/network connection	LED (green)
PS : Program status	LED (green/red)
0 ... 7 : Input signal status	LED (yellow)
F: Error status	LED (red)
$U_{LS} + U_A$: Supply status	LED (green)
SBM: Status of S-BUS Master	LED (green/red)
Indicators	Non-retentive
General information	
Dimensions (mm) W x H x L	75 x 35.7 x 117
Weight	Approx. 330 g

7 Mounting the Fieldbus Coupler

The fieldbus coupler can be fastened directly to your system using screws. It can also be mounted on a carrier rail using an adapter (WAGO accessory) or fastened to a profile rail using a surface mounting profile (WAGO accessory).

For mounting on a flat surface, WAGO offers spacers to assist in the mounting process that can be inserted between the 767 Series component. This helps by providing sufficient mounting distance for compact direct mounting, as well as eliminating gaps where dirt could accumulate. A cable tie can be fastened through each of two mounts in the spacer, which together serve to relieve strain from the sensor and actuator cables.

7.1 Information on Mounting

The following information shall always be observed:

- Disconnect the power supply from the system before you start with installation.
- The maximum diameter of the drill hole of the fieldbus coupler's mounting holes is not to exceed 4 mm. Otherwise, there may be no full contact with the fieldbus coupler's PE socket and a problem-free shielding is not possible.
- The cover cap of the DIP switch must be closed and bolted.
- To protect the fieldbus coupler from tensile forces that may arise, do not bridge spaces with it.
- Screw down the fieldbus coupler down only on flat contact surfaces to protect it from warping.
- Ensure that the connectors are not soiled during installation. Dirt and other such substances damage the contacts, allowing corrosion to develop.
- To avoid damaging the fieldbus coupler, do not mount it in shear areas of moving devices.
- Arrange for a sufficient potential equalization in your system.
- Use all mounting holes to mount the fieldbus coupler to your system so all FG (function ground) connections lie on a ground potential.

7.2 Tools and Accessories Required for Mounting

Depending on the mounting type, the following tools are required for installation:

- A screwdriver for M4 fixing screws
- Drilling machine to pre-drill the mounting holes for the fieldbus coupler to be mounted to the system and, if applicable, for the imperforated carrier rail.
- M4 thread cutter (bottoming tap or hand tap set)

The WAGO accessories listed below are required for mounting. The associated item numbers can be found in the "Accessories" section.

- Carrier rail adapter, including fixing screws and perforated or imperforated carrier rails (DIN 35 x 7.5) according to EN 60715, also available from WAGO.

or

- Profile adapter, including fixing screws
- Spacer (optional)

Three M4x12 screws are required for direct mounting of the fieldbus coupler. The length of the screw shaft is to be selected according to the mounting type.

Bore measurements

When fastening the 767 Series component without a threaded hole, the clearance hole must not be wider than 4mm so as to ensure safe contact of the FG (functional ground) connections.

7.3 Direct Mounting on Your System

Mount the fieldbus coupler directly on a level surface of your system, without using WAGO accessories. Direct mounting of the fieldbus coupler is to be carried out as follows:

1. Disconnect the power supply from those devices on which you wish to mount the fieldbus coupler.
2. Mark the drill holes using the hole drilling template printed on the packaging. You can also hold the fieldbus coupler in the desired position and mark the drill holes. Ensure that there is sufficient space around the 767 Series component to enable you to connect all cable without problems.

Note



We recommend using WAGO spacers for compact direct mounting. If these are used, the additional distance from the second 767 component is to be noted (see Section 7.7).

3. Fasten the fieldbus coupler to the grounded frame of your system or to another grounding point with the M4x12 screws via the three mounting holes.

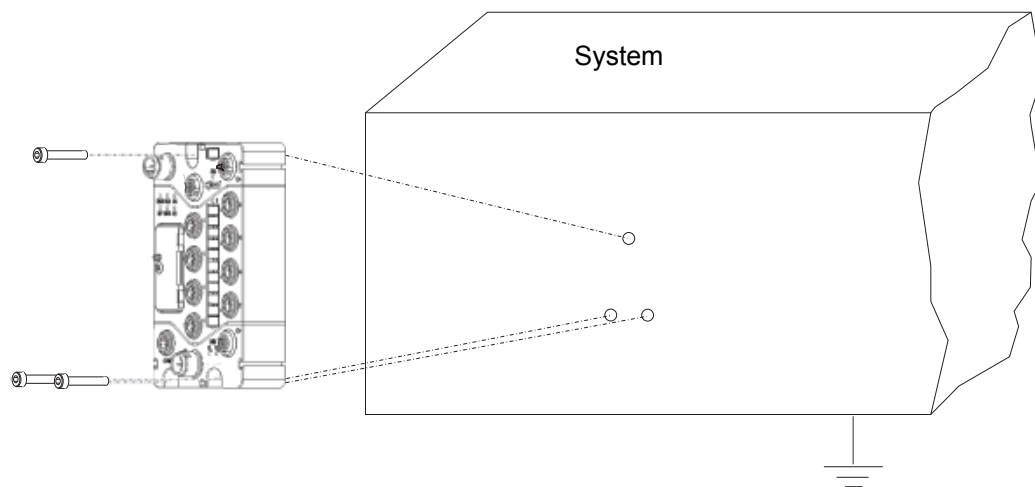


Figure 9: Mounting the fieldbus coupler on the grounded system

7.4 Mounting on a Carrier Rail (only with WAGO Accessories)

7.4.1 Fastening the Carrier Rail Adapter to the Fieldbus Coupler

A carrier rail adapter is required to mount the fieldbus coupler on carrier rails.

Screw together the fieldbus coupler and carrier rail adapter using the M4 threaded screws provided, as shown in the figure below.

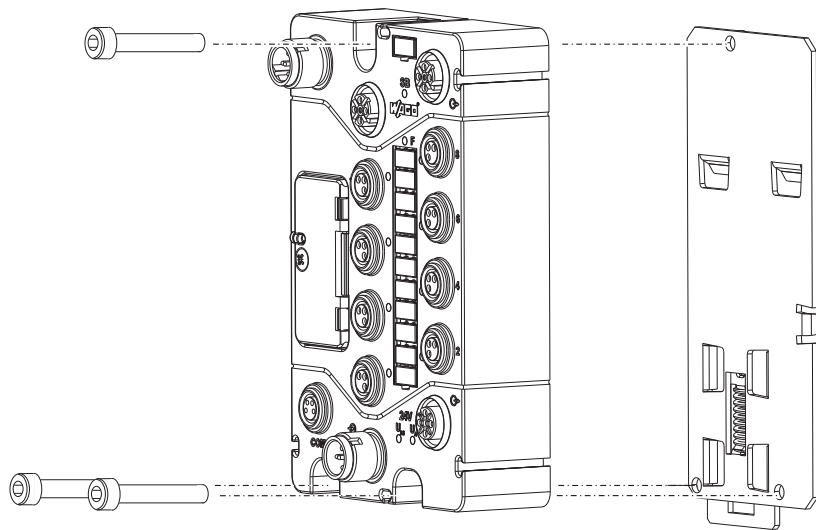


Figure 10: Fastening the fieldbus coupler to the carrier rail adapter

7.4.2 Fastening the Fieldbus Coupler with Carrier Rail Adapter to a Carrier Rail

In order to provide a clear representation, the carrier rail adapter in the figure below is shown without the fieldbus coupler.

When mounting the fieldbus coupler to a carrier rail (DIN rail 35 x 7.5) using a carrier rail adapter, proceed as follows:

1. Disconnect the power supply from those devices on which you wish to mount the fieldbus coupler.
2. Set the fieldbus coupler onto the edge of the carrier rail (61) with the two notches (60).
3. Press the undersurface against the lower carrier rail edge until the latch (62) locks in place.

Note



When mounting the rail vertically or if shock or vibration loading should occur, the use of end stops (item no.: 249-116 or 249-117) for stabilization is required. For more information, see the "Technical Data" section in the manual "WAGO SPEEDWAY 767, System Description and Notes."

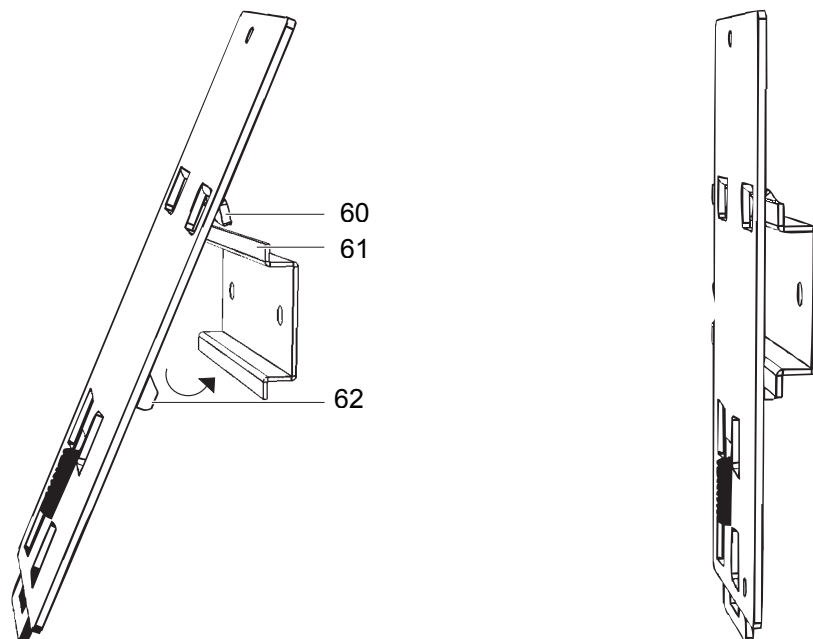


Figure 11: Mounting the carrier rail adapter

7.5 Mounting on a Profile Rail (only with WAGO Accessories)

7.5.1 Fastening the Profile Adapter to the Fieldbus Coupler

Aside from using carrier rail adapters to fasten the fieldbus coupler, you also have the option to fasten it to a profile rail using the profile adapter and nuts, provided that this mounting type is supported by your system. You are to supply the necessary nuts.

Screw together the fieldbus coupler and the profile adapter using the M4 threaded screws provided, as shown in the figure below.

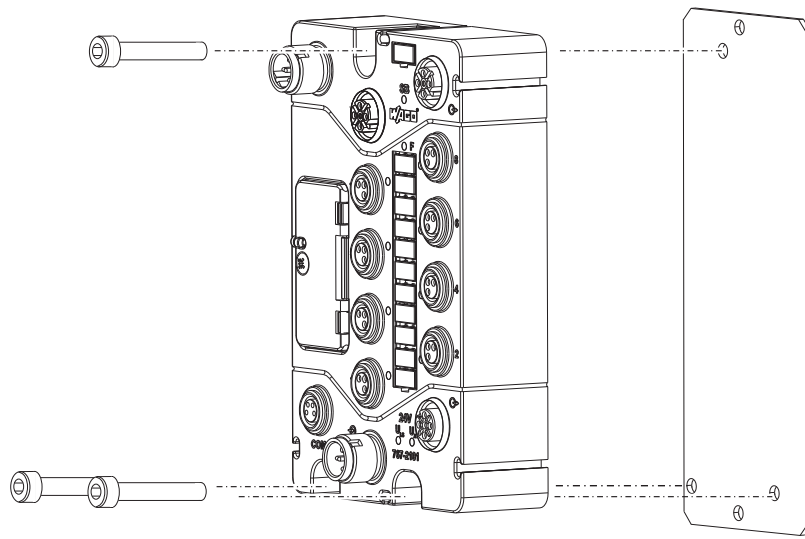


Figure 12: Fastening the fieldbus coupler to the profile adapter

7.5.2 Fastening the Fieldbus Coupler with Profile Adapter to a Profile Rail

To fasten the fieldbus coupler to a profile rail of your system, two nuts are required with one screw each (length of screw threads must be compatible with your system).

1. Disconnect the power supply from those devices on which you wish to mount the fieldbus coupler.
2. Insert the two screws into the holes above and beneath the fastened fieldbus coupler on the profile adapter.
3. Fasten an appropriate nut on each of these screws.
4. Insert the profile adapter with the attached fieldbus coupler into the profile rail of your system. Position it and tighten the screws.

7.6 Replacing the Marking Spaces

The module marker card and marking strip are attached when delivered. The protective cover is to be removed when labeling the marking strip. To do this, proceed as follows:

1. Press the slot screwdriver (maximum slot width: 3mm) into the small opening under the marking strip cover (12) and lever it up.
2. Remove the marking strip cover.
3. Mark the marking strip with a waterproof pen.
4. Reinsert the marking strip cover and press it firmly in place.

If the marker card (11) must be replaced, proceed in accordance with the step sequence described previously. New module marker cards can be obtained through WAGO.

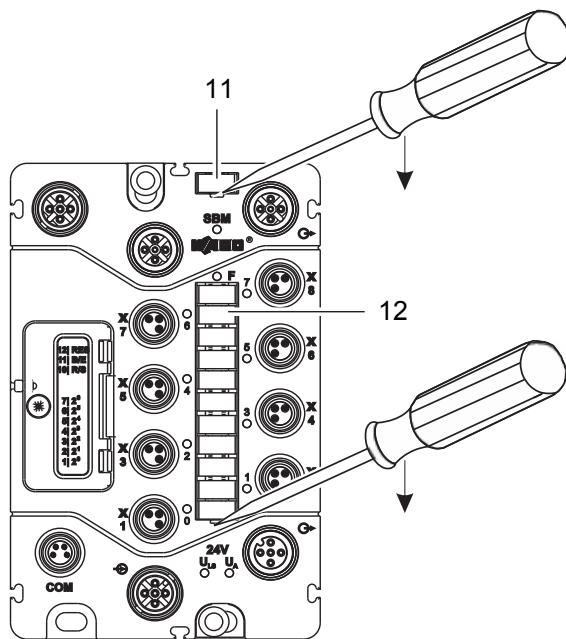


Figure 13: Replacing the marking spaces

7.7 Mounting the Spacer in the Case of Compact Arrangement

By using the spacer, a sufficient mounting distance can be achieved when directly mounting the 767 Series components, and gaps can be eliminated where otherwise dirt and other substances could accumulate. In addition, it is possible to optimize the cable routing of the sensors and actuators. For this purpose, two fastening lugs each are included on the spacer for cable ties.

1. Disconnect the power supply from those devices on which you wish to mount the fieldbus coupler.
2. The spacer can only be inserted into the appropriate openings of the fieldbus coupler from the bottom. To bind both components, place the fieldbus coupler on the spacer or push the spacer from the bottom into the coupler.
3. Fasten the components on a flat surface by fastening the fieldbus coupler to the grounded frame of your system or to another grounding point with three M4 screws via the mounting holes.

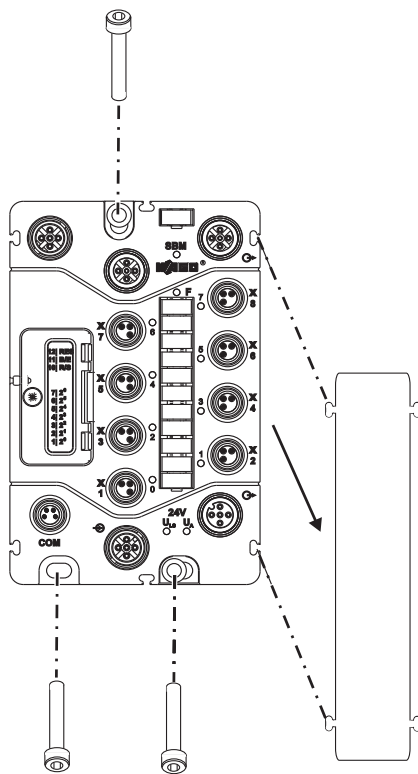


Figure 14: Attaching a spacer

4. When attaching 767 Series component, only one 767 component connected with a spacer can be attached and screwed on to the preceding component due to the mounting direction. The last 767 component is fastened without a spacer.

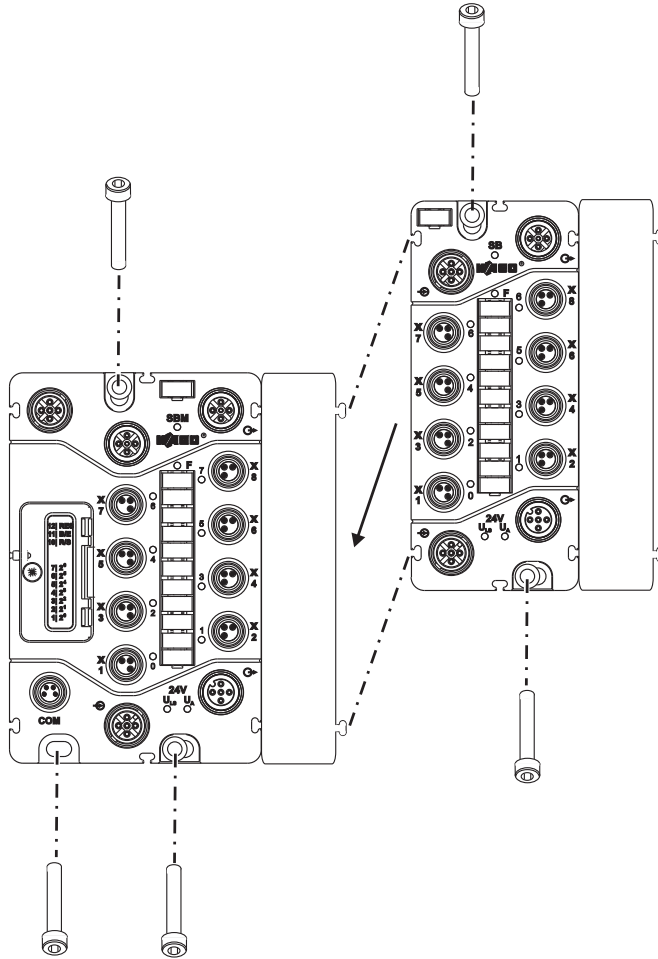


Figure 15: Attaching another 767 component to the fieldbus coupler

8 Connection to the Data and Supply Cable

8.1 Notes



DANGER

Electric voltage!

Operate the 767 Series components exclusively with 24VDC PELV (Protective Extra Low Voltage) or SELV (Safety Extra Low Voltage) voltage sources. Failure to comply may result in electric shock.

NOTICE

The highest current carrying capacity of the supply contacts is 4A!

Always observe the maximum current carrying capacity per supply line (U_{LS} , U_A) for each 767 component and the overall power consumption for all 767 Series component. Neither of these values shall exceed 4A since an increase in current causes the contacts to overheat and damages the 767 Series components. Information regarding the power demand of each 767 component can be found in the corresponding data sheet, which is available at www.wago.com.

NOTICE

Exposed connections!

If connections have not been closed with protective caps, liquid or dirt can penetrate the module and ruin it. Close all unused connections with protective caps to ensure that IP67 degree of protection is provided.

- The connectors must be disconnected from the power supply when screws are tightened.
- Tighten the connectors by hand only. Using mechanical tools can cause the threads to strip. In such a case, the fieldbus coupler must be replaced.

Tightening torques for the connectors: 0.6 Nm

- Observe the exact positioning (coding) between plug and socket.
- For both power supply and S-BUS, use only pre-assembled WAGO system cables so the specified characteristics of the technical data can be achieved.
- Keep all cables a sufficient distance away from electromagnetic sources of interference in order to maintain a high level of interference resistance of the 767 system against electromagnetic emissions.
- Observe the minimum bending radiuses of the WAGO system cable. For more information, see the technical data at www.wago.com.

- When laying all cable, ensure that you do not lay it in shear areas of moving machine parts.
- The cable shield of the fieldbus cable must be connected with functional ground on both ends over a wide area.
- Observe the correct layout of the potential equalization.
- Do not extend the fieldbus cable via the connectors.
- Do not use drop lines under any circumstances. This can lead to amplified line reflections and signal distortions, which greatly impair the transmission quality.

8.2 Required Accessories

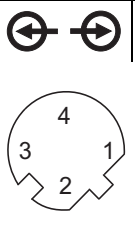
The WAGO accessories listed below are required for connecting the data and supply cable. The associated item numbers can be found in the "Accessories" section.

- S-BUS M12 terminator, IP 67
- S-BUS and supply cables, pre-assembled on both ends, IP 67
- USB cable pre-assembled on both ends, IP 67

8.3 Connecting the Ethernet Cable

The following table outlines the connection assignments for the Ethernet connections:

Table 5: Ethernet connection assignments

Connection	Contact	Description
	1	TX+
	2	RX+
	3	TX-
	4	RX-
	Connecting thread	Shielded

If you do not use a pre-assembled Ethernet cable, a shielded M12 socket with the IP 67 safety class is to be connected to this.

To connect a cable which is pre-assembled on one end only to an RJ45 plug, proceed as follows:

Contact	Connection	Color		Contact of RJ45 plug
1	TX+	Yellow	→	1
2	RX+	Blue	→	2
3	TX-	Brown	→	3
4	RX-	White	→	6

If the Auto MDI(X) functionality for the respective Ethernet connection FB1 or FB2 is activated, a cross-over cable is not necessary. This functionality automatically detects the direction for sending data and receiving data so that it is irrelevant which cable type is used (crossed or uncrossed). In the default status, the Auto MDI(X) functionality is activated. For more information, see Section 10.2.6.

8.3.1 Connecting a fieldbus coupler to an Ethernet network

To connect the fieldbus coupler to an Ethernet network, proceed as follows:

1. Disconnect the power supply from those devices on which you have mounted the fieldbus coupler.
2. Connect the fieldbus coupler with the Ethernet network by plugging the socket of the Ethernet cable (F) into the IN connection ⚡ (1).
3. Tighten the socket using the knurled-head screw.

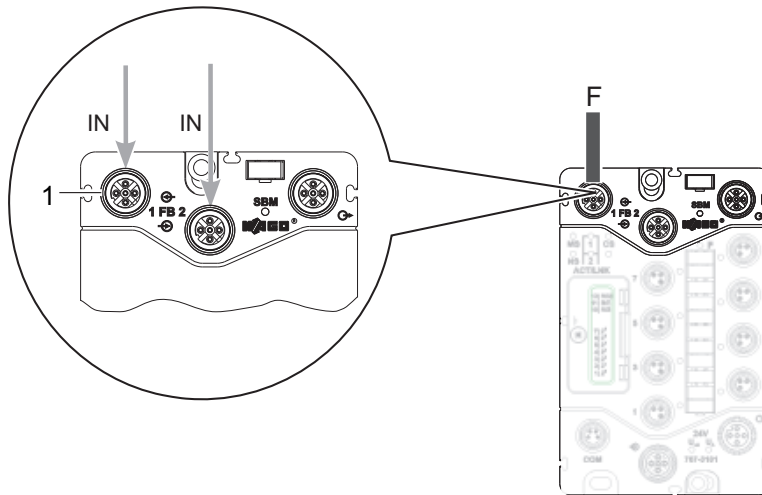


Figure 16: The fieldbus coupler is connected to the Ethernet network

8.3.2 Connecting several fieldbus couplers inside an Ethernet network

The network topology in the following figure consists of a mixed star and line topology. Up to 20 fieldbus couplers can be connected to the Ethernet network on the line topology. A switch is required to set up a star topology or a mixed topology. The number of fieldbus couplers on a star topology is limited only by the IEEE 802.3 Ethernet specification.

The connection of several fieldbus couplers with the Ethernet network can be carried out as shown in the following example:

1. Disconnect the power supply from those devices on which you have mounted the fieldbus coupler.
2. For the star topology, connect the respective Ethernet cable (W1, W2) with the IN connection  (1) of a fieldbus coupler and an Ethernet switch, as shown in the figure below. Tighten the sockets of the Ethernet cable.

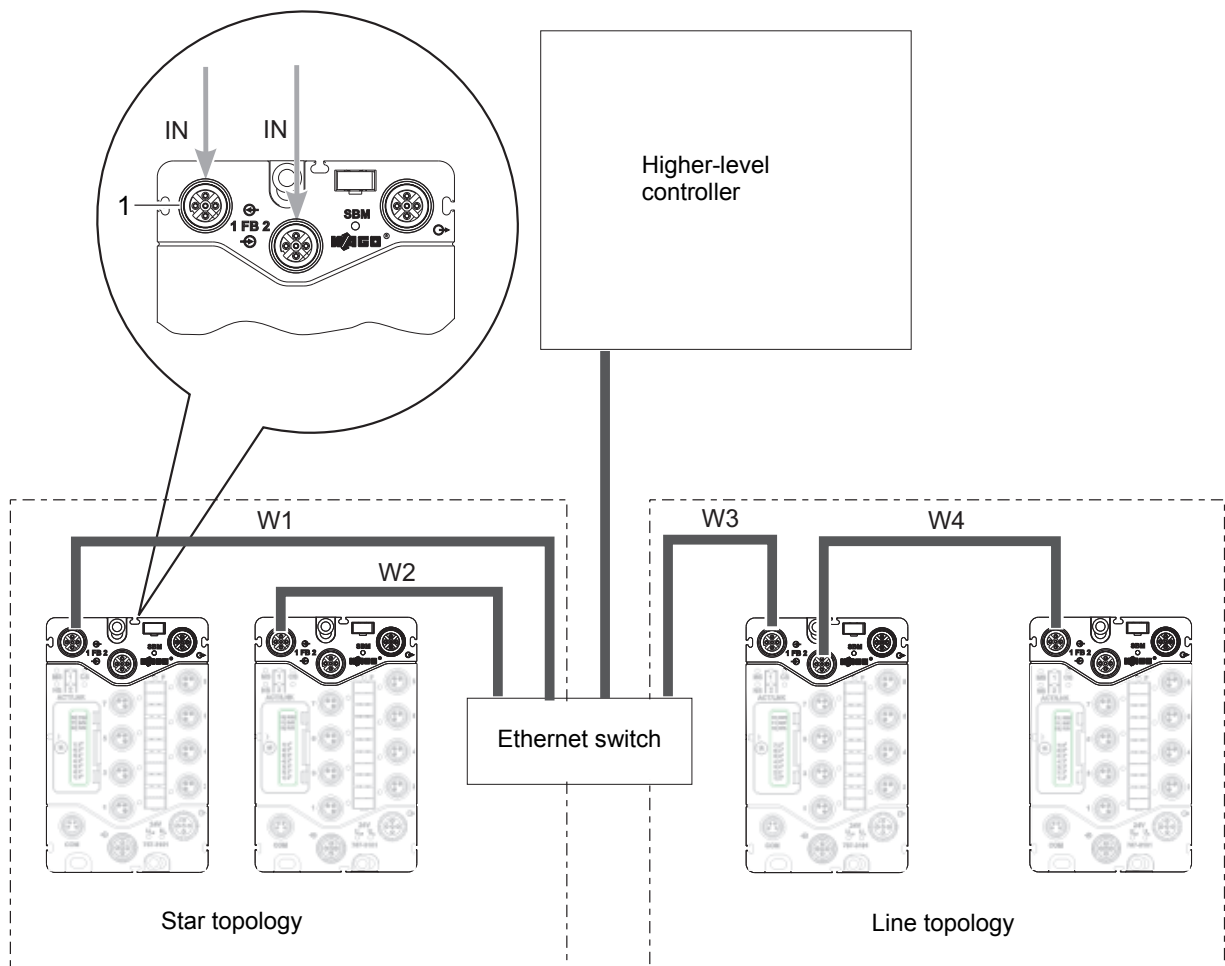


Figure 17: Example setup of a mixed topology with four fieldbus couplers

3. For a line topology, connect the Ethernet cable (W3, W4) with the connections IN $\oplus$ (1) and IN $\ominus$ (2) on the fieldbus coupler, as shown in the figure below. Tighten the sockets of the Ethernet cable.

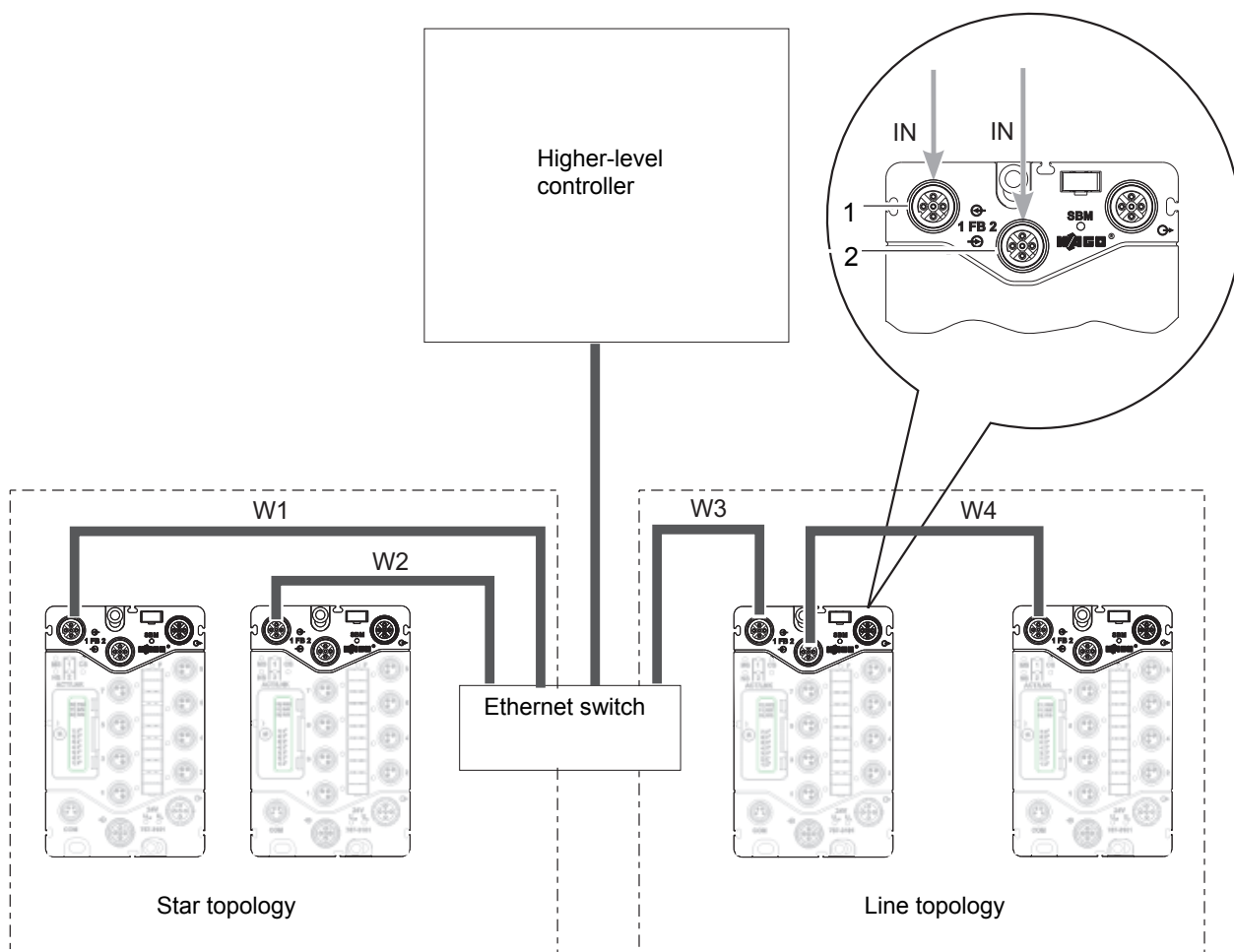


Figure 18: Example setup of a mixed topology with four fieldbus couplers

8.4 Connecting the S-BUS Cable

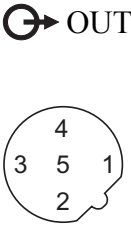
The S-BUS is used for communication between the fieldbus coupler and the connected I/O modules.

Requirement:

- A WAGO S-BUS cable pre-assembled on both ends is readily available to you. This is necessary for optimal signal transmission.
- The S-BUS terminator is available to you, which is necessary for communication.

The following table outlines the assignment of the S-BUS connection.

Table 6: S-BUS connection assignment

Connection	Contact	Description
	1	TD +
	2	TD -
	3	RD -
	4	RD +
	5	0VDC
	Connecting thread	Shielded

To connect the S-BUS cable to fieldbus couplers and I/O modules, proceed as follows:

1. Disconnect the power supply from those devices on which you have mounted the fieldbus coupler.
2. Connect the S-BUS cable (S1) with the OUT connection  (3) of the fieldbus coupler and the IN connection  (8) of the subsequent I/O module. For example, if two I/O modules have been connected to the fieldbus coupler, connect the S-BUS cables (S1, S2) to the associated IN and OUT connections, as shown in the figure below.
3. Tighten the plugs and sockets using the knurled-head screws.
4. Attach the S-BUS terminator (T) to the last I/O module as shown in the figure and tighten it.

If you do not connect any I/O modules to the fieldbus coupler, screw the S-BUS terminator (T) onto the S-BUS OUT output  (3) of the fieldbus coupler.

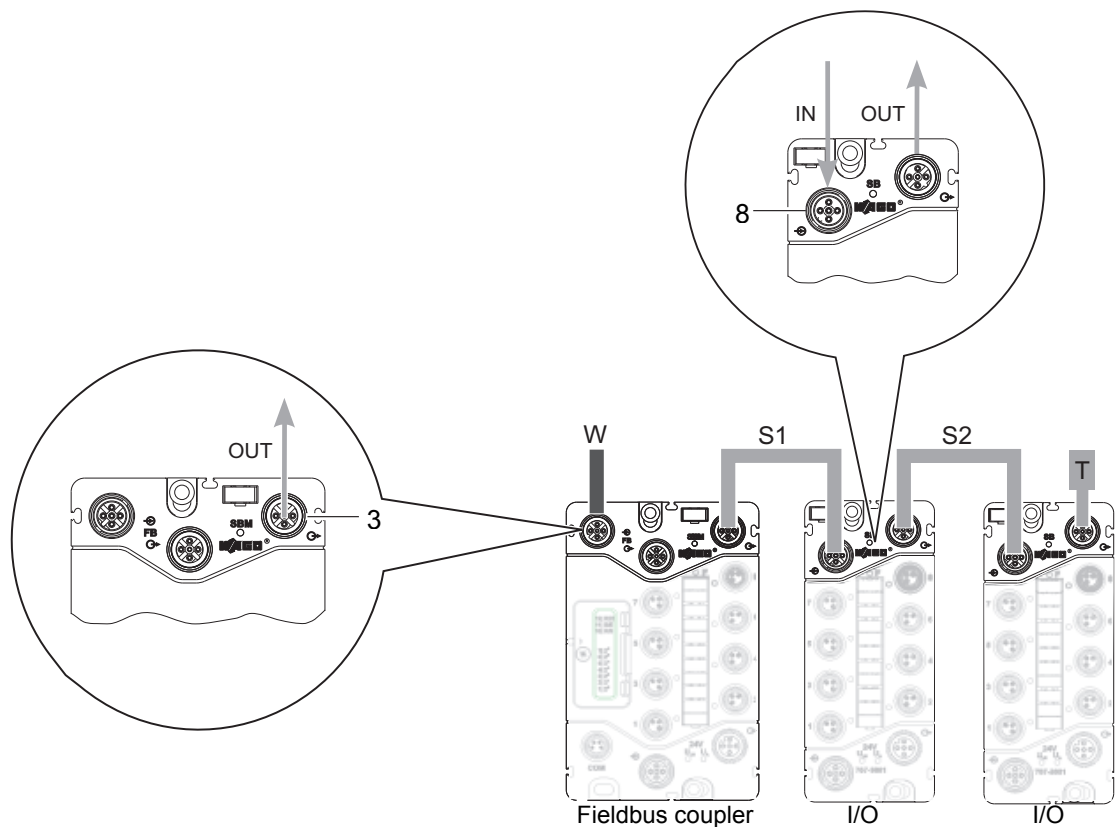


Figure 19: S-BUS is connected to one fieldbus coupler and two I/O modules

8.5 Connecting the Supply Cable

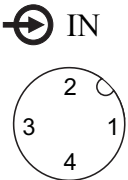
The supply cable provides power to the fieldbus coupler and the connected I/O modules.

Requirement:

- A supply cable for +24VDC and 0VDC must be connected to the supply input (6 in the figure on the next page) via appropriate fuses.
- The WAGO supply cable pre-assembled on both ends must be available (K1 and K2 in the figure on the next page).

The following table outlines the assignment of the supply connections:

Table 7: Supply connection assignment

Connection		Contact	Description
 IN	 OUT	1	24VDC U_{LS}
		2	24VDC U_A
		3	0V U_{LS}
		4	0V U_A

NOTICE

The highest current carrying capacity of the supply contacts is 4 A!

Always observe the maximum current carrying capacity per supply line (U_{LS} , U_A) for each 767 component and the overall power consumption for all 767 components. Neither of these values shall exceed 4A since an increase in current causes the contacts to overheat and damages the 767 components. Information regarding the power demand of each 767 component can be found in the corresponding data sheet, which can be downloaded at www.wago.com.

To connect the supply cable to the fieldbus coupler and I/O modules, proceed as follows:

1. Disconnect the power supply from those devices on which you have mounted the fieldbus coupler.
2. Connect the power supply cable (K0) with the fieldbus coupler by plugging the socket in the IN connection $\ominus$ (6) of the fieldbus coupler.
3. Tighten the socket using the knurled-head screw.
4. Connect the power supply transmission cable (K1) with the OUT connection $\oplus$ (5) of the fieldbus coupler and the IN connection $\ominus$ (8) of the subsequent I/O module.
For example, if two I/O modules have been connected to the fieldbus coupler, connect the power supply transmission cables (K1, K2) to the associated IN and OUT connections as shown in the figure below.
5. Tighten the plugs and sockets using the knurled-head screw.

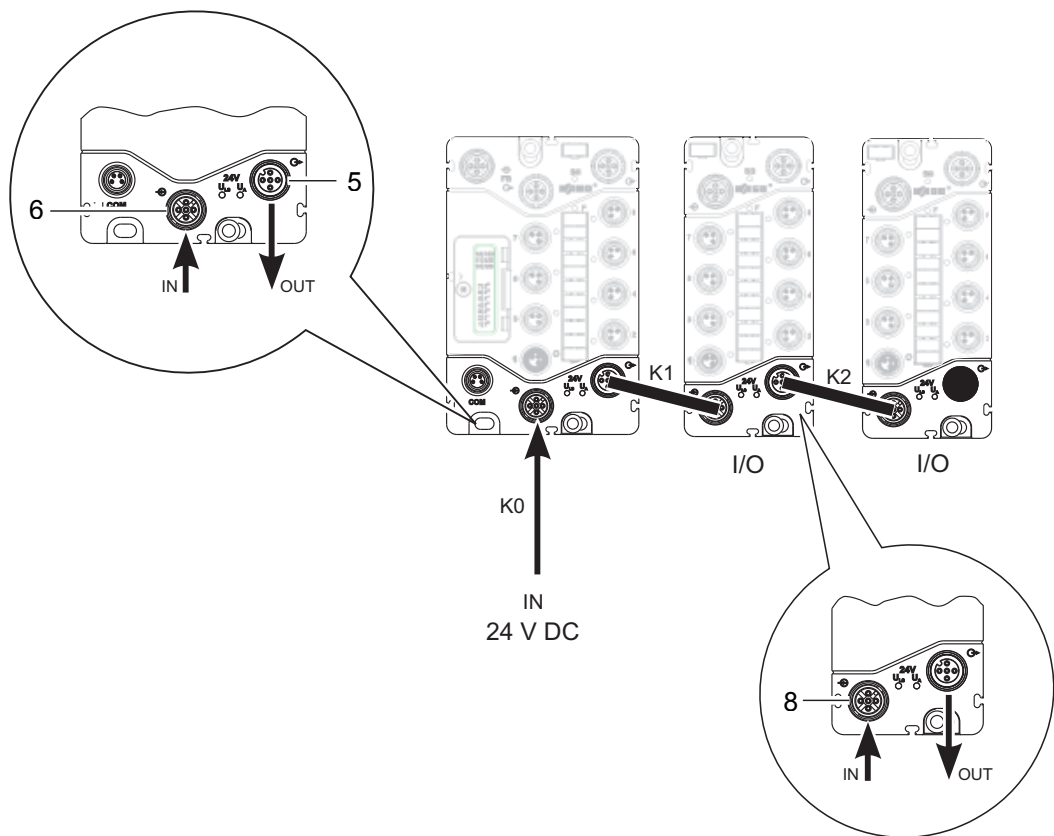


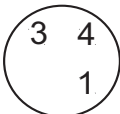
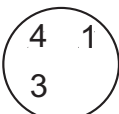
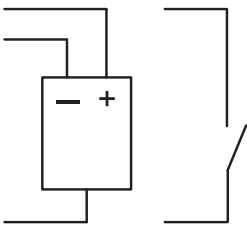
Figure 20: Supply cable connected with fieldbus coupler and I/O modules

8.6 Connecting the Sensor Cable

The sensor cables provide power to the connected sensors and transmit the sensor signals.

When using cables that have not been pre-assembled, make sure that these cables are equipped with M8 plugs rated to IP67. The following table outlines the assignment of the sensor connections:

Table 8: Connection assignment of digital inputs

Connection		Connection diagramm
IN	IN	1: 24 V 3: 0 V U_{LS}
 X1, X3, X5, X7	 X2, X4, X6, X8	 4: Input

NOTICE

The highest current carrying capacity of the supply contacts is 4A!

Ensure that the sensors from the U_{LS} supply line are supplied with power. The sensor's power consumption is to be taken into consideration when determining the present power demand for the U_{LS} supply line.

NOTICE

The total maximum power consumption of the sensors must not exceed 400mA (50mA/channel).

Please note that the combined power consumption of all connected sensors is not to exceed 400mA. The distribution of power among the existing connections is depending on the individual power requirements of the sensors.

To connect the sensors to the digital inputs (X1 – X8), proceed as follows:

1. Disconnect the power supply from those devices on which you have mounted the fieldbus coupler.
2. Insert the sensor cable plug into a digital input socket (4) of the fieldbus coupler, and tighten it via knurled-head screw.
3. Screw a protective cap on all unused ports to ensure that IP67 degree of protection is provided.

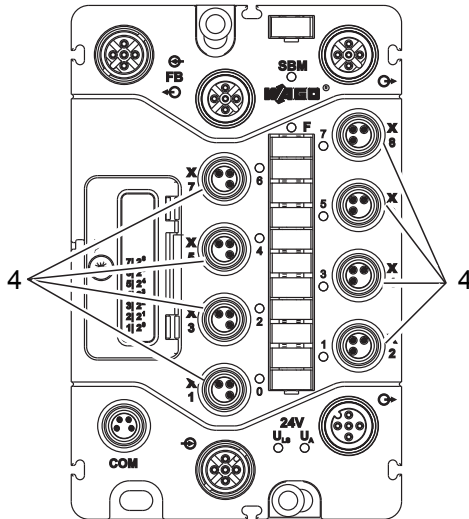


Figure 21: Digital inputs

8.7 Connecting the USB Cable

The fieldbus coupler's USB connection can be used to either configure the fieldbus coupler with an FDT/DTM frame application or to use general service functions. To use the USB connection, the USB driver "WAGO_767_USB_DRIVER_Setup" (item no.: 759-922) must be installed on your computer. This driver can be obtained on the Internet at www.wago.com. The USB driver is also included on the CD-ROMs "WAGOframe" (item no.: 759-370) and „CoDeSys 3“ (item no.: 759-915).

After installing the WAGO USB driver, the following COM ports are supplied by the USB connection:

- **I/O-Service**
This COM port is needed to configure the fieldbus coupler using WAGOframe.
- **I/O-Programming**
This COM port is required for serial communication between the CoDeSys 3 development environment and the fieldbus coupler.

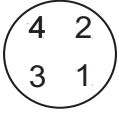
Note



When using the USB connection, a faulty USB data transmission or an outage can cause communication disruptions. This can be due to additional or inappropriate USB distributors (hubs) or to USB cables that are too long (max. 5 m) or inappropriate. One should therefore refrain from using additional devices if possible.

When using a USB cable that has not been pre-assembled, M8 plugs with the IP 67 degree of protection are to be connected. The following table outlines the assignment of the USB connection:

Table 9: USB connection assignment

Connection	Contact	Description
IN 	1	+ 5V
	2	- Data
	3	+ Data
	4	0VDC
	Connecting thread	Shielded

1. Connect your PC with the fieldbus coupler by inserting the USB cable plug into the fieldbus coupler's USB connection COM (7) and tightening it.
2. Switch on the fieldbus coupler. Please refer to Section 9.6 for more information. After the fieldbus coupler has been switched on, your operating system detects a new device and installs the USB driver.

Note



During installation of the driver, an MS Windows message may appear several times informing you that the Windows Logo Test failed. Ignore this message and click **[Continue with installation]**.

3. When the USB connection is not in use, screw a protective cap on it to comply with the IP 67 degree of protection.

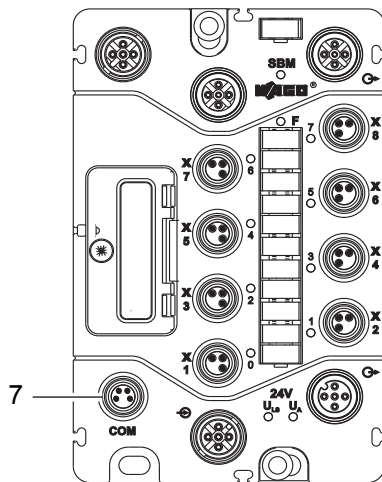


Figure 22: USB connection

9 Start-Up

NOTICE

Exposed connections!

If connections have not been closed with protective caps, liquid or dirt can penetrate the module and ruin it. Close all unused connections with protective caps to ensure that IP67 degree of protection is provided.

Before starting up the 767 node, ensure that the following requirements are met:

- The fieldbus coupler is mounted properly (see Section 7).
- All necessary supply and sensor lines, as well as the S-BUS and Ethernet cable, are securely fastened onto the appropriate connections (see Section 8).
- The S-BUS terminator is fastened (see Section 8).
- The power supply of the higher-level controller is switched on (see associated manual).
- An appropriate potential equalization is implemented in your system.
- The shielding is carried out properly.

9.1 Determining the IP Address of the Host PC

For the host PC to be able to communicate with the fieldbus coupler in the Ethernet network, both must be in the same subnet*.

Example:

Your host PC has the IP address 192.168.1.1 and the fieldbus coupler should be in the same subnet. For this to be the case, the first three figures of the PC and the fieldbus coupler must agree in the subsequent network mask (with a period separating each): **255.255.255.0**

Host PC

IP Address	192.168.1.1
Network mask	255.255.255.0

Subnet address range for the fieldbus coupler

192.168.1.2 ... 192.168.1.254

To determine the IP address of the host PC (with the MS Windows operating system) using the MS DOS prompt, proceed as follows.

1. Click on "Start" and select "Execute".
2. Enter the command `cmd` and press the Enter key. A prompt opens.
3. Enter the command `ipconfig` and press the Enter key.
4. The IP address, subnet mask and standard gateway, including the appropriate parameters, are displayed.
5. Now select an IP address for your fieldbus node which must be in the same network as your PC.
6. Take note of the selected IP address:

IP address of your fieldbus node: ----- . ----- . ----- . -----

* A subnet is a logical division of a network.

9.2 Assignment of an IP Address

To set up a communication between a PC and the fieldbus coupler, each device must be in the position to send data to the other. So that the data can arrive at the right place, the receiver must be clearly specified (addressed). This takes place in IP networks with an IP address.

The fieldbus coupler can obtain its IP address dynamically (from a server) or it can be configured with a static (fixed) IP address.

The dynamic assignment of the IP address can be carried out via the BootP or DHCP protocol in the web-based management or in WAGOframe. Dynamically assigned IP addresses are not saved however. Therefore, a BootP or DHCP server must be available for each restart.

You can configure an IP address via

- the DIP switch on the fieldbus coupler,
- the web-based management (WBM) or
- the "WAGOframe" configuration software.

To access the web-based management (WBM) in a browser, the fieldbus coupler requires an IP address. This initial IP address can be assigned via the DIP switch using WAGOframe.



Note

In the default status, the assignment of dynamic IP addresses is activated via the BootP protocol.

9.3 Assignment an IP Address using BOOTP

In the default status, IP address assignment is activated via the BootP protocol. Assigning the IP address using the BootP protocol is illustrated here with the example of the WAGO BootP server.

Requirement:

The WAGO BootP server must be installed on your PC. You can obtain the BootP server on the "Tools & Docs" CD or on the WAGO website.

Configuring the WAGO BootP server

In order to communicate with the fieldbus coupler in the local network, the MAC address of the fieldbus coupler must first be specified and an IP address must then be assigned.

1. Take note of the MAC address (51) of the fieldbus coupler. This can be found on the label on the side of the fieldbus coupler.

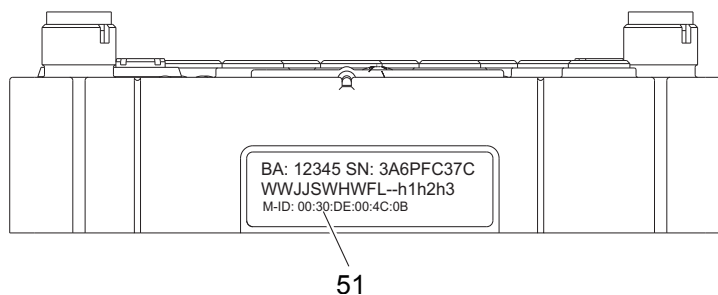


Figure 23: Example of a MAC address on the adhesive label on the left side of the fieldbus coupler

2. Turn on your PC.
3. In the "Start" menu, start the WAGO BootP server by navigating to **Programs > WAGO Software > WAGO BOOTP Server**.
4. Open the configuration file in which you are assigning the MAC address of an IP address from the same network. To do this, click **[Edit BootPtab]** in the BootP server. The configuration file will then be opened.

5. Click on the selected line of the configuration file:

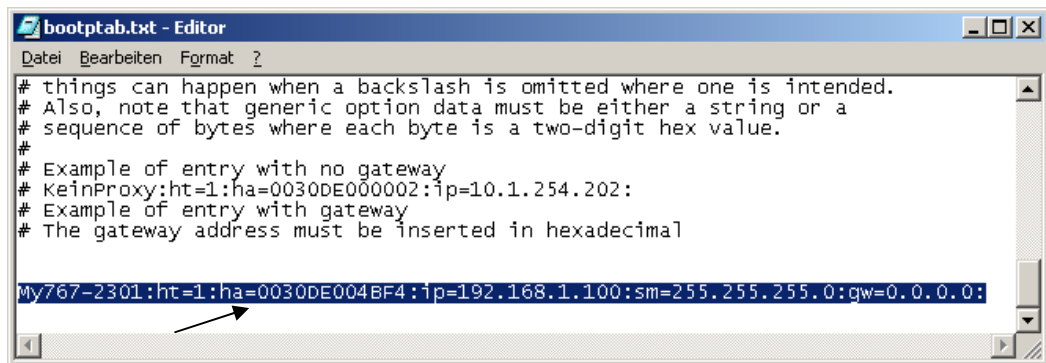


Figure 24: Configuration file

6. Replace the 12-character MAC address after "ha=" (see arrow) with the address printed on the label (51).
7. Enter an IP address after "ip=". In the example shown here, this is 192.168.1.100. Separate the individual numbers with a decimal point.
8. To address more than one fieldbus node, add a line for each node with the corresponding assignment for each additional fieldbus coupler to the file bootptab.txt. Repeat steps 5 to 7 for this.
9. Save the new settings in the bootptab.txt file. To do this, click on "File" in the menu and select "Save".
10. Close the editor.

Table 10: Explanations of the configuration line

Parameter	Description
Node_1	Arbitrary name of Ethernet node.
ht=1	Network hardware type. For Ethernet, this is 1.
ha=0030DE000200	MAC ID of adhesive label on the side of the fieldbus coupler.
ip= 10.1.254.100 ip= 10.1.254.200	IP address for the fieldbus coupler which is connected to the same network as the host PC.
gw=10.1.136.1	IP address for the gateway. For a local network, you do not need to specify a gateway.
Sm=255.255.0.0	Subnet mask of the subnet to which the fieldbus coupler should belong.

Assigning an IP address using the WAGO BootP server

1. To start the BootP server click the button **[Start]** in the BootP dialog window that is opened. Various messages will be displayed in the window. The error messages indicate that some services (e.g. port 67, port 68) in the operating system have not been defined. You can ignore these error messages.

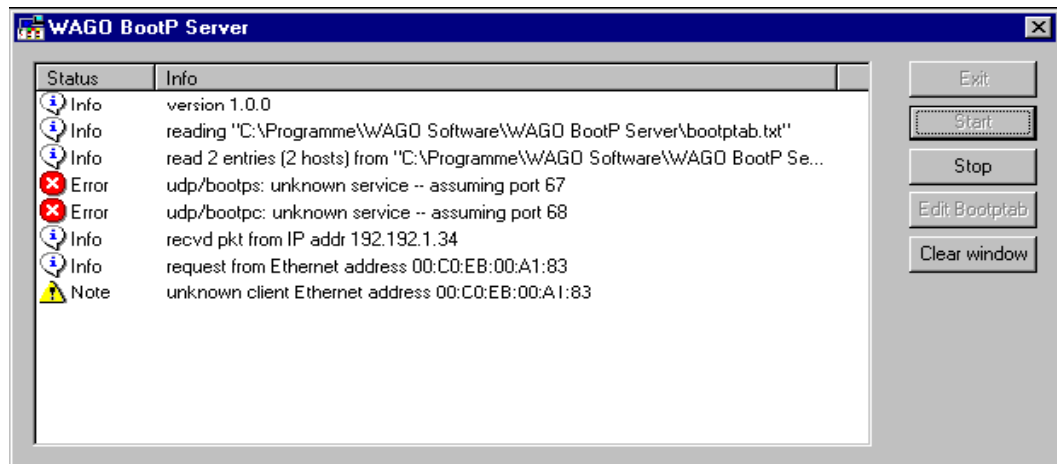


Figure 25: Dialog window of the WAGO BootP server with messages

2. Restart the fieldbus coupler by disconnecting it from the power supply and then reconnecting it.
A request appears from the fieldbus coupler. The BootP server replies that the IP address has been accepted (no errors). The IP address is now temporarily available in the fieldbus coupler, but is not yet permanently saved. When restarted, the fieldbus coupler attempts to obtain a new IP address from the BootP server.
3. Click **[Stop]** and then **[Exit]** to close the BootP Server.
4. To permanently save the IP address in the fieldbus coupler, select the option "Static IP" in the "TCP/IP view of the web-based management. Or use WAGOframe by selecting "Device memory" under the "IP address obtained from" parameter.

9.4 Setting a Permanent IP Address using DIP Switches

To integrate the fieldbus coupler in the Ethernet network, the fieldbus coupler must be assigned an IP address that is unique across the network. This can be done using DIP switches. The address can be any number between 1 and 254. In the default status, the top three bytes (network ID) are assigned "192.168.1." and the subnet mask 255.255.255.0. Use switches 1 – 8 to set the last byte (host ID) of the IP address. To do this, switch 9 (Set Network) must be set to "On".

Table 11: Default settings of DIP switch

Switches	1	2	3	4	5	6	7	8	9	10	11	12
Binary values	2^0 (1)	2^1 (2)	2^2 (4)	2^3 (8)	2^4 (16)	2^5 (32)	2^6 (64)	2^7 (128)	Set Network	Run/ Stop	Boot/ Execute	Reset
Switch setting	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off	Off

Requirement:

The fieldbus coupler must not be connected to the power supply.

If you were to set the IP address 192.168.1.100, you would proceed as follows:

1. Open the cover by unscrewing the M3 screw with a screwdriver.
2. Set switch 9 (Set Network) to "On" to activate an IP address by means of the DIP switch.
3. For the host ID 100, set switches 3, 6 and 7 to "On".
($2^2 + 2^5 + 2^6 = 4 + 32 + 64 = 100$). Please see the figure on the next page.



Note

With the 8 available switches you can set addresses from 1 – 254. Addresses 0 and 255 are reserved and may not be used.

4. Close the cover and screw it securely in place to maintain degree of protection IP 67.



Note

The new IP address is only active when the fieldbus coupler is connected to the power supply.

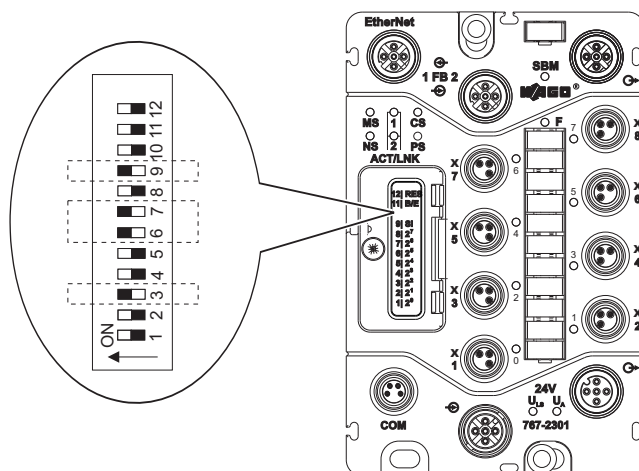


Figure 26: DIP set for the host ID 100

9.4.1 Assigning an IP Address using the Web-Based Management

Detailed information on this subject can be found in Section 10.2.2.

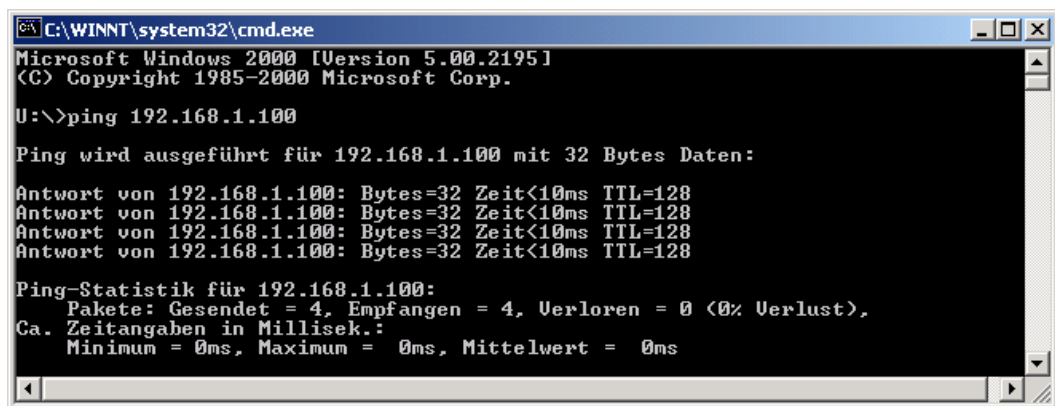
9.4.2 Assigning an IP Address using WAGOframe

Detailed information on this subject can be found in Section 16.

9.5 Testing the Network Connection

Carry out a ping network function to check whether the fieldbus coupler can be reached at the IP address you have assigned in the network. To do this, open the prompt in MS Windows by clicking "Start" and selecting **Programs > Run**. Enter cmd in the dialog box and click **[OK]**.

1. In the DOS window, enter the command ping and the IP address of the fieldbus coupler; e.g.: ping 192.168.1.100.
2. Press the Enter key. Your PC sends out an inquiry that is answered by the fieldbus coupler. This reply appears in the DOS window. If the error message "Timeout" appears, the fieldbus coupler has not responded properly. If this occurs, please check your network settings.



```
C:\WINNT\system32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

U:\>ping 192.168.1.100

Ping wird ausgeführt für 192.168.1.100 mit 32 Bytes Daten:

Antwort von 192.168.1.100: Bytes=32 Zeit<10ms TTL=128
Antwort von 192.168.1.100: Bytes=32 Zeit<10ms TTL=128
Antwort von 192.168.1.100: Bytes=32 Zeit<10ms TTL=128
Antwort von 192.168.1.100: Bytes=32 Zeit<10ms TTL=128

Ping-Statistik für 192.168.1.100:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
```

Figure 27: Example of a fieldbus node function test

3. If the test is completed successfully, close the DOS window.

9.6 Switching on the Fieldbus Coupler

After you have set the IP address on the fieldbus coupler, turn on the power supply. The fieldbus coupler and all connected I/O modules are ready for operation after the initialization phase.



Note

No 767 components are to be added or removed during operation, as this causes a disruption in the 767 node.

10 Configuration

10.1 Access Configuration of Output Modules

The write access to the output module is only possible via MODBUS or EtherNet/IP. The unique assignment is configured using WAGOframe or the control. This is possible for every I/O module.

The MODBUS protocol for the fieldbus coupler is factory activated (MODBUS TCP, MODBUS UDP). Thus, the output modules can only be modified using MODBUS.

Note



If you configure the output module in CoDeSys, the outputs of the controller are assigned independent of the previous configuration of other users.

If only a fieldbus is activated (MODBUS or EtherNet/IP), this fieldbus is automatically assigned all output modules. Activation and deactivation are possible via the web-based management or WAGOframe.

The following table gives an example of which fieldbus has write access. MODBUS is considered active when either MODBUS/TCP or MODBUS/UDP or both are activated. The activation steps are executed in ascending order (1 – 4).

Table 12: Example of access configuration for MODBUS and EtherNet/IP

No.	Status of activation	Write access
1	MODBUS TCP: activated MODBUS UDP: activated EtherNet/IP: deactivated	MODBUS
2	MODBUS TCP: activated → deactivated MODBUS UDP: activated EtherNet/IP: deactivated → activated	MODBUS Because no fieldbus was exclusively activated, the write access remains with MODBUS.
3	MODBUS TCP: deactivated MODBUS UDP: activated → deactivated EtherNet/IP: activated	EtherNet/IP
4	MODBUS TCP: deactivated MODBUS UDP: deactivated → activated EtherNet/IP: activated	EtherNet/IP Because no fieldbus was exclusively activated, the write access remains with EtherNet/IP.

10.2 The Web-Based Management (WBM)

The implemented websites of the web-based management help with configuring the fieldbus coupler. To access these, open your browser and enter **http://** into the address line, followed by the IP address of the fieldbus coupler (e.g. **http://192.168.1.100**). Please refer to section 9.2. to assign an IP address to the fieldbus coupler.

The WBM is password-protected; therefore, you will be prompted to enter a password after selecting a link from the navigation bar. The user names and passwords are explained in Section 11.1.

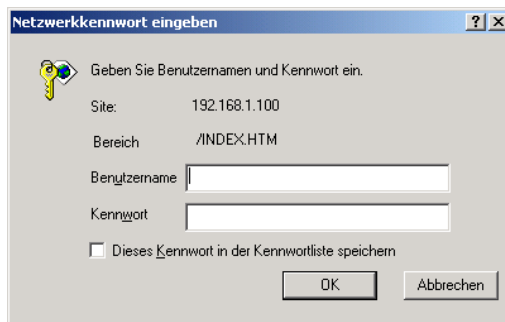


Figure 28: "Enter network password" dialog box

10.2.1 "Information" View

After entering the IP address, the start page "Information" appears from the web-based management. This site provides information on the fieldbus coupler, the network and the blink code.



WAGO
INNOVATIVE CONNECTIONS

Web Based Management

Navigation

- **Information**
- TCP/IP
- Port
- Watchdog
- Clock
- Ethernet
- Users
- Administration

Status Information

Coupler Details

Order Number	0767-2301
Mac Address	00:30:DE:00:00:00
Firmware Revision	01.01.04 (00)

Network Details

IP address	192.168.1.3
Subnet mask	255.255.0.0
Gateway	0.0.0.0
Hostname	0030DE000000
Domainname	wago.com

Module Status

State Modbus Watchdog TCP:	Disabled
State Modbus Watchdog UDP:	Disabled
Error Group	0
Error Code	0
Error Argument	0
Error Description	Coupler running, OK

Figure 29: "Information" view

The following table explains the parameters listed on the site:

Table 13: Description of the "Information" view parameters

Coupler details	
Order number	Displays the item number of the fieldbus coupler
Mac address	Displays the Mac address, which helps identify and address the fieldbus coupler
Firmware revision	Displays the current status of the firmware
Network details	
IP address	Displays the current IP address of the fieldbus coupler
Subnet mask	Displays the current subnet mask of the fieldbus coupler
Gateway	Displays the current gateway address of the fieldbus coupler
Host name	Displays the current host name
Domain name	Displays the current domain name

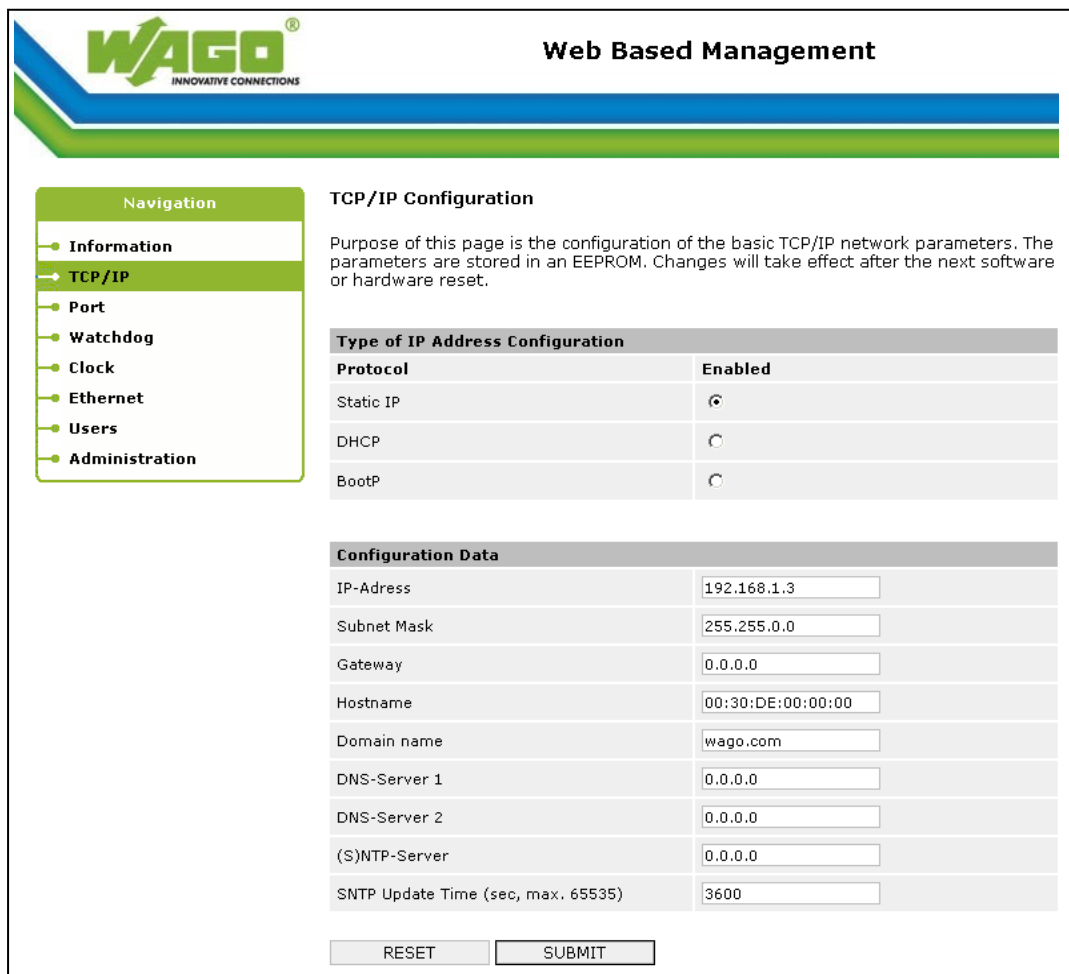
Table 13: Description of the "Information" view parameters

Module status	
MODBUS Watchdog TCP State	Displays whether the Modbus TCP Watchdog is activated. Disabled: not active Running: active Expired
MODBUS Watchdog UDP State	Displays whether the Modbus UDP Watchdog is activated. Disabled: not active Running: active Expired
Error group	The error group, error code and error argument all show the arising error. Please see Section 15.3.3.
Error code	
Error argument	
Error description	Description of the error.

10.2.2 "TCP/IP" View

The parameters for the Ethernet configuration can be changed on this TCP/IP configuration site.

1. To confirm your entries, click **[SUBMIT]**.
2. To clear your entries, click **[RESET]**.
3. Your entries are not active until the software has been reset. To reset the software, click on "Administration" in the navigation bar and then "Software reset". You can also disconnect the power supply from the fieldbus coupler and then reconnect it.



WAGO
INNOVATIVE CONNECTIONS

Web Based Management

Navigation

- Information
- TCP/IP**
- Port
- Watchdog
- Clock
- Ethernet
- Users
- Administration

TCP/IP Configuration

Purpose of this page is the configuration of the basic TCP/IP network parameters. The parameters are stored in an EEPROM. Changes will take effect after the next software or hardware reset.

Type of IP Address Configuration	
Protocol	Enabled
Static IP	☒
DHCP	☐
BootP	☐

Configuration Data	
IP-Address	192.168.1.3
Subnet Mask	255.255.0.0
Gateway	0.0.0.0
Hostname	00:30:DE:00:00:00
Domain name	wago.com
DNS-Server 1	0.0.0.0
DNS-Server 2	0.0.0.0
(S)NTP-Server	0.0.0.0
SNTP Update Time (sec, max. 65535)	3600

RESET **SUBMIT**

Figure 30: "TCP/IP" view

The following table explains the parameters listed on the site:

Table 14: Description of the "TCP/IP" view parameters

Type of IP address configuration	
Static IP	Selection option of static or dynamic IP addressing
DHCP (dynamic)	
BootP (dynamic)	
Configuration data	
IP address	Here you can set a static IP address through which the fieldbus coupler is to be accessed. This is active when "Static IP" is activated in the "Type of IP address configuration" field.
Subnet mask	Set the subnet mask here. This is active when "Static IP" is activated in the "Type of IP address configuration" field.
Gateway	Set the gateway here. This is active when "Static IP" is activated in the "Type of IP address configuration" field.
Domain name	Set the domain name here.
DNS Server 1	Set the address of the DNS server here.
DNS Server 2	Set the address of an alternative DNS server here.
(S)NTP Server	Set the IP address of the time server here.
SNTP Update Time (sec, max. 65535)	Set the cycle time here to specify how often the time should be requested from the time server.

10.2.3 "Port" View

On this protocol configuration page, you can select the protocols that you desire for communication.

1. To confirm your entries, click **[SUBMIT]**.
2. To clear your entries, click **[RESET]**.
3. Your entries are not active until the software has been reset. To reset the software, click on "Administration" in the navigation bar and then "Software reset". You can also disconnect the power supply from the fieldbus coupler and then reconnect it.

WAGO
INNOVATIVE CONNECTIONS

Web Based Management

Navigation

- Information
- TCP/IP
- Port**
- Watchdog
- Clock
- Ethernet
- Users
- Administration

Port Configuration

Purpose of this page is the configuration of the network protocols. The configuration data is stored in an EEPROM. Changes will take effect after the next software or hardware reset.

Protocol	Port	Enabled
FTP	21	☒
SNTP	123	☐
HTTP	80	☒
Ethernet IP	44818 (TCP), 2222 (UDP)	☐
Modbus UDP	502	☒
Modbus TCP	502	☒

RESET SUBMIT

Figure 31: "Port" view

The following table explains the parameters listed on the site:

Table 15: Description of the "Port" view parameters

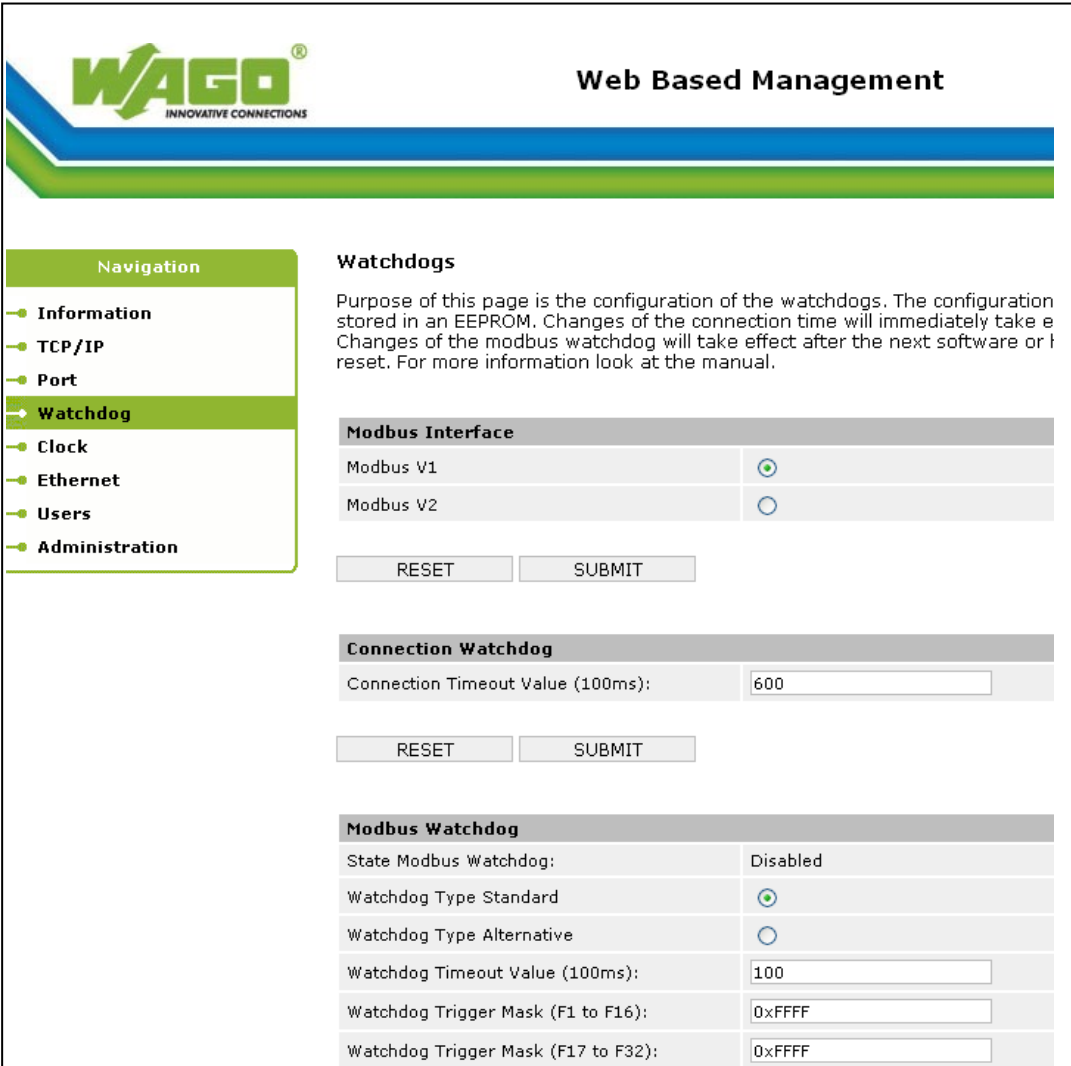
Port settings	
FTP	When this checkbox is selected, the protocol for file transfer is active.
SNTP	When this checkbox is selected, the time is updated using Ethernet.
HTTP	When this checkbox is selected, the web server (WBM) is active.
Ethernet IP	When this checkbox is selected, the protocol for exchanging process data using EtherNet/IP is active.
MODBUS UDP	When this checkbox is selected, the protocol for exchanging process data using MODBUS UDP is active.
MODBUS TCP	When this checkbox is selected, the protocol for exchanging process data using MODBUS TCP is active.

10.2.4 "Watchdog" View

On this page, the MODBUS Watchdogs can be configured.

1. To confirm your entries, click **[SUBMIT]**.
2. To clear the entries that have not yet been saved, click **[RESET]**.
3. Your entries up until the time of connection are not active until the software has been reset. To reset the software, click on "Administration" in the navigation bar and then **[Software Reset]**. You can also disconnect the power supply from the fieldbus coupler and then reconnect it.

MODBUS V1



WAGO
INNOVATIVE CONNECTIONS

Web Based Management

Navigation

- Information
- TCP/IP
- Port
- Watchdog**
- Clock
- Ethernet
- Users
- Administration

Watchdogs

Purpose of this page is the configuration of the watchdogs. The configuration stored in an EEPROM. Changes of the connection time will immediately take effect. Changes of the modbus watchdog will take effect after the next software or hardware reset. For more information look at the manual.

Modbus Interface

Modbus V1	☒
Modbus V2	☐

RESET SUBMIT

Connection Watchdog

Connection Timeout Value (100ms):

RESET SUBMIT

Modbus Watchdog

State Modbus Watchdog:	Disabled
Watchdog Type Standard	☒
Watchdog Type Alternative	☐
Watchdog Timeout Value (100ms):	100
Watchdog Trigger Mask (F1 to F16):	0xFFFF
Watchdog Trigger Mask (F17 to F32):	0xFFFF

Figure 32: "Watchdog" view

The following table explains the parameters listed on the site:

Table 16: Description of the "Watchdog" view parameters

MODBUS Interface	
MODBUS V1	Standard MODBUS implementation (preset). When selecting V2, the MODBUS, the setting options change.
MODBUS V2	767 MODBUS implementation. When selecting V1, the MODBUS, the setting options change.
Connection Watchdog	
Connection timeout value (100 ms)	The connection watchdog monitors the activity of all TCP connections. If no communication takes place during the parameterized time, the connection is closed and the resources are released.
MODBUS Watchdog	
State MODBUS Watchdog	Actual watchdog states Disabled: not active (0x0000) Running: active (0x0001) Expired (0x0002)
Watchdog Type Standard	Non-latching watchdog that has to be stopped after tripping from a MODBUS register before it starts again.
Watchdog Type Alternative	Latching watchdog that is started again with the first valid MODBUS TCP command.
Watchdog Timeout Value (1000 ms)	Specifies in what intervals "MODBUS Requests" are expected from a disconnected MODBUS Master.
Watchdog Trigger Mask (F1 to F16)	Default setting: 0x0000 Use this entry field to set the function codes (1 – 16) to trigger the watchdog function. The function code can be selected with "1". (= $2^{(\text{function code}-1)} + 2^{(\text{function code}-1)} + \dots$) Example: Function code 5 $\rightarrow 2^{(5-1)} = 2^4 \rightarrow$ bit 4 is set to 1 A value not equal to zero starts the watchdog function. If only codes from unsupported functions are entered in the screen, the watchdog does not start. An existing error is reset and the process image can be described again. Again here, no changes can be made when the watchdog is running. When the watchdog is active, there is no code by which
Watchdog Trigger Mask (F17 to F32)	Default setting: 0x0000 The same function as before, but with the function codes 17 to 32. Therefore, this register should be left at the default value.

MODBUS V2

WAGO
INNOVATIVE CONNECTIONS

Web Based Management

Navigation

- Information
- TCP/IP
- Port
- Watchdog**
- Clock
- Ethernet
- Users
- Administration

Watchdogs

Purpose of this page is the configuration of the watchdogs. The configuration stored in an EEPROM. Changes of the connection time will immediately take effect. Changes of the modbus watchdog will take effect after the next software or hardware reset. For more information look at the manual.

Modbus Interface

Modbus V1	☐
Modbus V2	☒

RESET SUBMIT

Connection Watchdog

Connection Timeout Value (100ms):

RESET SUBMIT

Modbus Watchdog TCP

State Modbus Watchdog:	Disabled
Watchdog Timeout Value (100ms):	0
Watchdog Configuration:	0x0000

RESET SUBMIT

Modbus Watchdog UDP

State Modbus Watchdog:	Disabled
Watchdog Timeout Value (100ms):	0
Watchdog Configuration:	0x0000

Figure 33: "Watchdog" view 2

Table 17: Description of the "Watchdog" view parameters

MODBUS Interface	MODBUS Interface
Connection Watchdog	
Connection Timeout Value (100 ms)	The connection watchdog monitors the activity of all TCP connections. If no communication takes place during the parameterized time, the connection is closed and the resources are released.
MODBUS Watchdog TCP/UDP	
State MODBUS Watchdog	Actual watchdog states Disabled: not active (0x0000) Running: active (0x0001) Expired (0x0002)
Watchdog Timeout Value (100 ms)	Specifies in what intervals "MODBUS Requests" are expected from a disconnected MODBUS Master.
Watchdog Configuration	Set the properties of the watchdog here.

10.2.5 "Clock" View

On this page, the real-time clock is configured.

1. To confirm your entries, click **[SUBMIT]**.
2. To clear your entries, click **[RESET]**.
3. Your entries are not active until the software has been reset. To reset the software, click on "Administration" in the navigation bar and then "Software reset".
You can also disconnect the power supply from the fieldbus coupler and then reconnect it.

Figure 34: "Clock" view

The following table explains the parameters listed on the site:

Table 18: Description of the "Clock" view parameters

Configuration data	
Time on Device	Set the time here.
Date (YYYY-MM-DD)	Set the date here.
Time Zone (+/- hour:minute)	Set the time zone here.
Summer Time	If this checkbox is activated, the WBM is set to summer time.
Display mode	
12-Hour Clock	Switching between 12-hour and 24-hour time display.

10.2.6 "Ethernet" View

On this Ethernet configuration page, you can select the transfer rates and diverse switch settings that you wish to use. The fieldbus coupler has three ports for which you can carry out various settings: two outer ports (Ethernet connections FB1 and FB2) and one internal Ethernet switch.

1. To confirm your entries, click **[SUBMIT]**.
2. To clear your entries, click **[RESET]**.
3. Your entries are not active until the software has been reset. To reset the software, click on "Administration" in the navigation bar and then "Software reset".
You can also disconnect the power supply from the fieldbus coupler and then reconnect it.

WAGO
INNOVATIVE CONNECTIONS

Web Based Management

Navigation

- Information
- TCP/IP
- Port
- Watchdog
- Clock
- Ethernet**
- Users
- Administration

Ethernet Configuration

Purpose of this page is the configuration of the Ethernet Switch settings. The new configuration is stored in an EEPROM. Changes will take effect after the next softw or hardware reset.

Speed Configuration		
Desc	Port 1	Port 2
Enable Port	☒	☒
Enable Power-Save Mode	☒	☒
Enable Auto MDI-X	☒	☒
Enable Autonegotiation	☐	☐
10 MBit Half Duplex	☐	☐
10 MBit Full Duplex	☐	☐
100 MBit Half Duplex	☐	☐
100 MBit Full Duplex	☐	☐
Enable Full Duplex Flow Control	☒	☒

RESET SUBMIT

Bandwidth and Sniffer Configuration			
Desc	Port 1	Port 2	Port 3
Limit Mode	All	All	All
Input Limit Rate	No Limit	No Limit	No Limit
Output Limit Rate	No Limit	No Limit	No Limit
Sniffer Port	☐	☐	☐
Receive Sniff	☐	☐	☐
Transmit Sniff	☐	☐	☐

BC = Broadcast, MC = Multicast, FU = Flooded Unicast

RESET SUBMIT

Figure 35: "Ethernet" view

The following table explains the parameters listed on the site:

Table 19: Description of the "Ethernet" view parameters

Speed configuration	
Enable Port	You can switch the ports on and off. Checkbox selected: port switched on Checkbox unselected: port switched off
Enable Power-Save Mode	If the port is not used, the power-save mode is activated.
Enable Auto MDI-X	Activates the Auto MDI-X function of the Ethernet port. This function employs an internal switching functionality to allow the use of either crossed (cross-over) or straight (patch) cable.
Enable Autonegotiation	If this radio button is selected, the automatic search for optimal Ethernet speed is activated.
10 Mbit Half Duplex	If this radio button is selected, the fixed Ethernet speed is activated.
10 Mbit Full Duplex	
100 Mbit Half Duplex	
100 Mbit Full Duplex	
Enable Full Duplex Flow Control	If there is a connection to a switch which also supports "Flow Control", both devices can agree on a disruption of the flow of data if needed. Checkbox selected: switched on Checkbox unselected: switched off
Bandwidth and sniffer configuration	
Limit Mode	This selection box allows you to define which types of messages are restricted. All: Restriction of all Ethernet messages BC: Broadcast messages MC: Multicast messages FU: Flooded Unicast messages
Input Limit Rate	This selection box allows you to set the restriction of the transfer rate for the input data.
Output Limit Rate	This selection box allows you to set the restriction of the transfer rate for the output data.
Sniffer Port	TCP packets from other ports are forwarded to this port. This function assists in analyzing the network.
Receive Sniff	All incoming packets are forwarded to the sniffer port.
Transmit Sniff	All outgoing packets are forwarded to the sniffer port.

10.2.7 "Users" View

On this page, the passwords of the users **admin**, **user** and **guest** can be changed. Passwords can only be changed by the **admin** user. An overview of passwords can be found in Section 11.1.

1. From the "user" selection box, select the user whose password you wish to change.
2. Enter your new password in the "New password" field.
3. Reenter the password in the "Confirm password" field to verify spelling.
4. To confirm your entries, click **[SUBMIT]**.
5. To clear your entries, click **[RESET]**. The password of the selected user is changed.

WAGO
INNOVATIVE CONNECTIONS

Web Based Management

Navigation

- Information
- TCP/IP
- Port
- Watchdog
- Clock
- Ethernet
- Users**
- Administration

Users

Purpose of this page is to change the passwords for the existing users. Changes will take effect immediately.

The access is only allowed for admins.

Authorization

Admin-Password:

Configuration Data

User:

New Password:

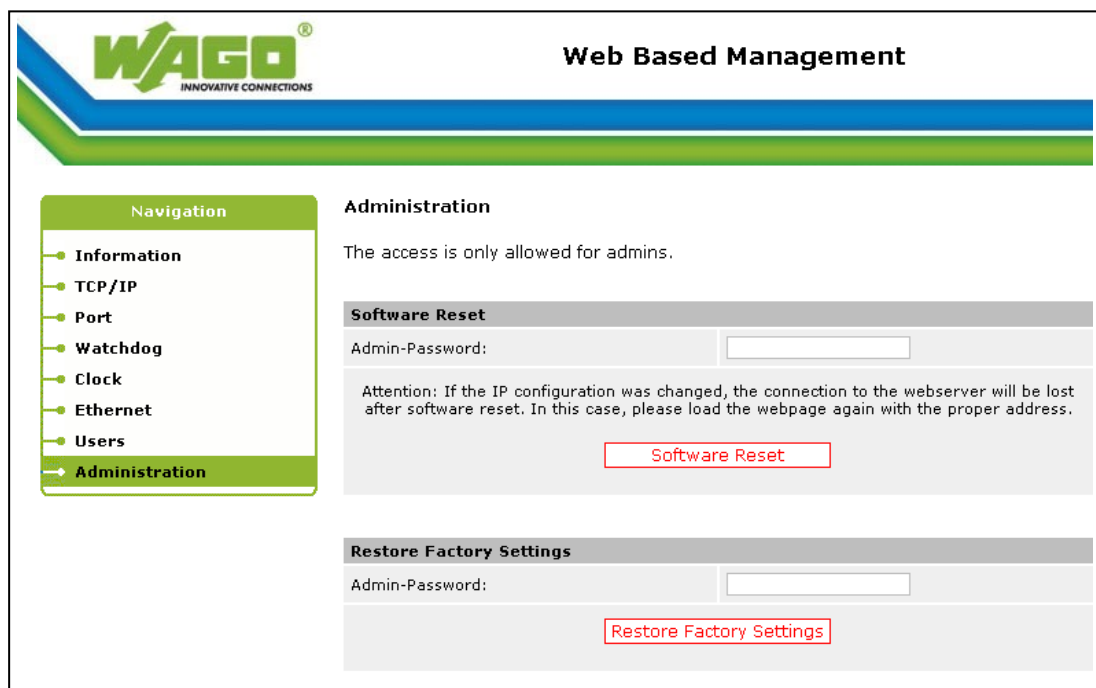
Confirm Password:

Figure 36: "Users" view

10.2.8 "Administration" View

On this page, you can reset the software or reset all parameters to the default status. You can only use these functions if you are logged on as the **admin** user. An overview of the passwords can be found in Section 11.1.

1. To activate modified parameters in the fieldbus coupler, click on **[Reset Software]**.
2. To reset all parameters of the WBM to the default status, click on **[Restore Factory Settings]**. This also resets the IP address, causing the connection to the fieldbus coupler to be lost. To assign a permanent IP address, set the DIP switch 9 to "On"; to assign an IP address using BootP, DHCP, etc., set it to "Off".



WAGO
INNOVATIVE CONNECTIONS

Web Based Management

Navigation

- Information
- TCP/IP
- Port
- Watchdog
- Clock
- Ethernet
- Users
- Administration**

Administration

The access is only allowed for admins.

Software Reset

Admin-Password:

Attention: If the IP configuration was changed, the connection to the webserver will be lost after software reset. In this case, please load the webpage again with the proper address.

Software Reset

Restore Factory Settings

Admin-Password:

Restore Factory Settings

Figure 37: "Administration" view

11 File System

Three partitions of the file system are available to the user including a RAM disk and two partitions in Flash memory.

Table 20: Summary of file system partitions

Partition	Format	Type	Size	Use
R:\	FAT 12	RAM disk	1MB	-
U:\	FAT 12	Flash disk	1MB	WBM pages
P:\			1MB	CoDeSys programm

The R:\ partition is designed as nonpermanent RAM disk and can be used to temporarily store data. These files are lost when the system is restarted. To save files permanently, use partitions "U:\ and P:\.

The WBM sites are located in "U:\". Partition "P:\ " can be accessed by FTP and also directly from the CoDeSys development environment.

11.1 User Administration

Access to the web-based management and via FTP is password protected. In the default status, the following users and access authorizations are defined:

Table 21: User administration

User name	Password	WBM	FTP	WAGOframe
admin	wago	Full access	Full access	Full access
user	user	Limited access	Full access	Limited access
guest	guest	Limited access	Full access	Read access

Use WBM and WAGOframe to change the passwords.

11.2 Access via FTP

To access the fieldbus coupler using FTP, proceed as follows:

1. Open the DOS console. Click on the start menu and select "Run". Enter the cmd command and click [OK].
2. In the DOS console, enter the ftp command with the IP address of the fieldbus coupler. In the following example, the command is
ftp 192.168.1.100
3. To log on to FTP, enter your user name and password.

You are now logged on in FTP. You can now, for example, access the directory with the pages of the web-based management by entering the `cd U:\` command. By means of the regular FTP commands, you have the option of copying these pages to your PC and customizing them.

```

Auswählen C:\WINNT\system32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

U:\>ftp 192.168.1.100
Verbindung mit 192.168.1.100 wurde hergestellt.
220 Nucleus FTP Server (Version 1.?) ready.
Benutzer (192.168.1.100:(none)): admin
331 User name okay, need password.
Kennwort:
230 User logged in, proceed.
ftp> cd U:\
250 Requested file action okay, completed.
ftp> cd private
250 Requested file action okay, completed.
ftp> dir
200 Command okay.
150 File status okay; about to open data connection.
drw-rw-rw-  1 owner  group           0 Jan 21 13:38 .
drw-rw-rw-  1 owner  group           0 Jan 21 13:38 ..
drw-rw-rw-  1 owner  group           0 Jan 21 13:38 images
-rw-rw-rw-  1 owner  group    3201 Jan 21 13:38 tcpip.ssi
-rw-rw-rw-  1 owner  group    5257 Jan 21 13:38 styles.css
-rw-rw-rw-  1 owner  group    2484 Jan 21 13:38 state.ssi
-rw-rw-rw-  1 owner  group    3883 Jan 21 13:38 port.ssi
-rw-rw-rw-  1 owner  group    1993 Jan 21 13:38 clock.ssi
-rw-rw-rw-  1 owner  group     679 Jan 21 13:38 index.htm
226 Closing data connection. Requested file action successful.
FTP: 612 Bytes empfangen in 0,00Sekunden 612000,00KB/s
ftp> quit
221 Service closing control connection.  Logged out if appropriate.

U:\>

```

Figure 38: Connection setup using FTP in the DOS console

12 EtherNet/IP

12.1 EDS File

The EDS file contains the characteristics of the fieldbus coupler and information regarding its communication capabilities. The EDS file is imported and installed by the corresponding configuration software.

The EDS file can be obtained at www.wago.com. When installing the EDS file, please refer to the information provided in the documentation of the configuration software which you are using.

12.2 Process Image

After starting up the fieldbus coupler, it automatically identifies all connected I/O modules. The fieldbus coupler uses this to create a local process image, divided into an input and output data range.

If you have selected the fieldbus EtherNet/IP for the fieldbus coupler, the process image is structured with word alignment. Data larger than one byte is displayed internally in Intel format ("Little Endian"). If both analog and digital I/O modules are plugged in to the node, the digital input and output data are combined byte by byte and attached behind the analog data in the process image. Fieldbus variables that can be processed via an IEC 61131 program are attached behind the process image of the I/O modules.

The size of the process image of an Ethernet node can be identified from the connected 767 components. The process image has a total size of 4096 bytes: 2048 bytes of input data and 2048 bytes of output data. In addition, 512 bytes of fieldbus variables are available in the process image for input data and output data respectively.



Note

The input and output process images can accommodate more data than your higher-level controller can transmit. If the data is larger than the telegram of the higher-level controller, use the assembly instances to select which data should be transmitted.

You can obtain access to the process image (including the fieldbus variables) via the assembly instances or directly via the EtherNet/IP objects. Please see Sections 12.6 and 12.7 for more information.

12.3 Process Data Exchange

The assembly instances are required in order to exchange process data between the higher-level controller and the fieldbus coupler. Using these assembly instances, you can write and read the inputs and outputs of the 767 components.

12.3.1 Assembly Instances

A process data exchange is initiated using the assembly instances. From the existing static assembly instances (see Section 12.6.3), select one instance for each sending direction.

For transmission of process data from the higher-level controller (originator) to the fieldbus coupler (target), the following assembly instances are available:

- Instance 101: for analog and digital output data
- Instance 102: for digital output data
- Instance 103: for analog output data
- Instance 111: for fieldbus input variables
- Instance 113: for analog and digital output data with diagnostic acknowledgement

For transmission of process data from the fieldbus coupler to the higher-level controller, the following assembly instances are available:

- Instance 104: for analog and digital input data, including status byte
- Instance 105: for digital input data, including status byte
- Instance 106: for analog input data, including status byte
- Instance 107: for analog and digital input data
- Instance 108: for digital input data
- Instance 109: for analog input data
- Instance 110: for fieldbus coupler output variables
- Instance 112: for analog and digital input data with status byte, current error and diagnosis



Note

If you select an assembly instance without status information (107 – 110), information such as fieldbus errors, pending diagnosis and process data status (valid/invalid) are not available and cannot be integrated into the application of the higher-level controller. The status byte is described in class 64_{hex}, attribute 5.

12.3.2 Identifying the Data Length

After you have selected the necessary assembly instance, you need to identify the length of the data in each sending direction of the respective instance for the configuration software of the higher-level controller. You have the following options:

Reading out the lengths via the assembly instances

The simplest option involves reading out the instance attribute 4 from each assembly instance between 101 and 113 which should be used. Attribute 4 of the assembly classes (ID = 4) indicates the total number of data bytes transmitted in this direction. This number includes the overall size of the respective input or output data, including status byte, error, diagnosis and fieldbus variables. The individual data sizes are not identified by reading out the assembly instances. This is only possible using the instance attribute of class 64_{hex}.

Reading out the lengths via class 64_{hex}, instance 1 (see Section 18.1.1)

The bit lengths of the analog and digital I/O modules connected to the Ethernet node are read out via the instance attributes 7 – 10. Moreover, the instance attributes 100 – 105 are used to specify the number and offset of fieldbus variables that are added to the overall process image.

Depending on which assembly instance you select for exchanging the input and output data, the status byte is also to be added, as well as three bytes for the current error and/or diagnosis. The lengths of the diagnostic data is made up of a fixed quantity of 10 bytes and a variable quantity. The variable quantity is dependent on the I/O modules connected to the 767 node. This quantity of bytes can be read out via attribute 53.

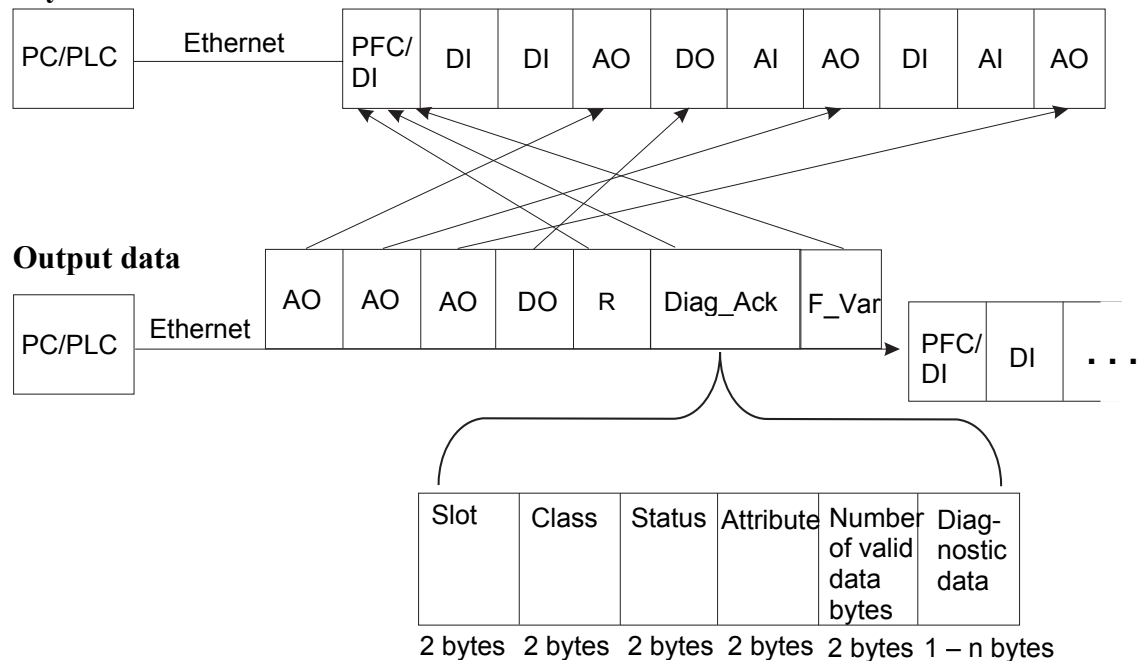
The fieldbus variables are then attached to the process image. The length of the fieldbus variables can be calculated using the instance attributes 100 – 105. Detailed information regarding the instance attributes can be found in the corresponding table of class 64_{hex}.

12.3.3 Example with Assembly Instances 101 and 104

In this example, the assembly instance 101 is used for exchanging data between the higher-level controller and the fieldbus coupler and includes all analog and digital output data. The assembly instance 104, which includes the status byte in addition to the analog and digital input data, is used for exchanging data between the fieldbus coupler and the higher-level controller.

A sample configuration results in the following structure of process data which is sent from the higher-level controller to the fieldbus coupler:

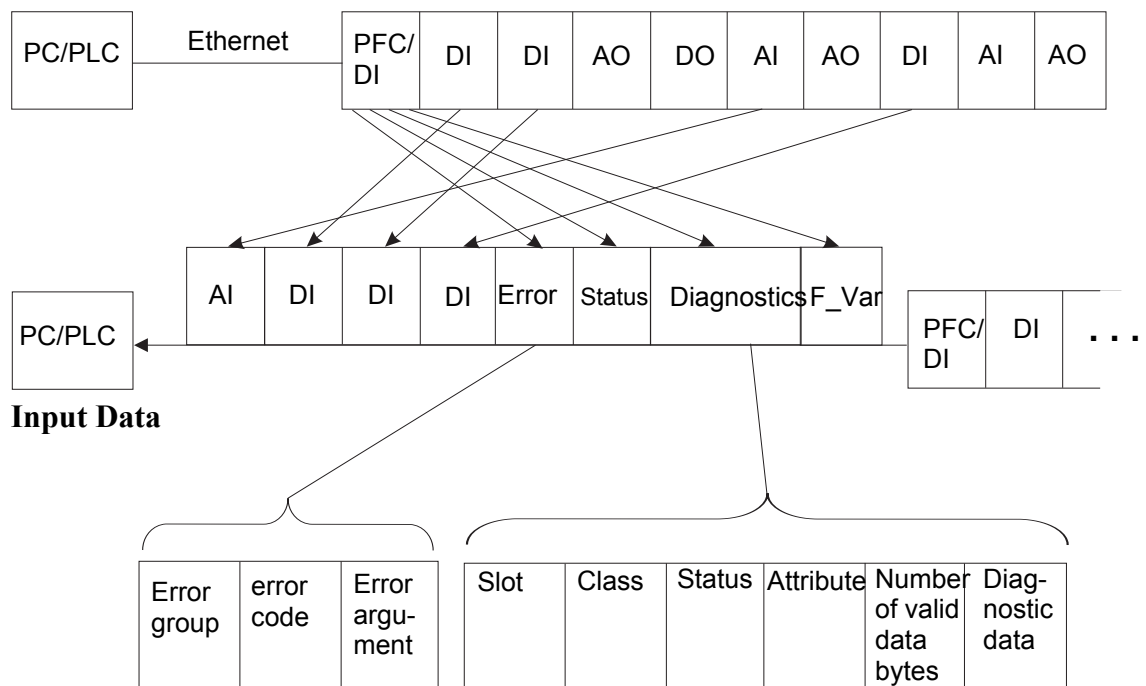
Physical structure



PFC/DI: Digital inputs on the fieldbus coupler
 DI: Digital input, 8 bits
 DO: Digital output, 8 bits
 AI: Analog input, 8 bytes
 AO: Analog output, 8 bytes
 F_Var: Fieldbus input variables, 12 bytes
 Diag_Ack: Diagnostic acknowledgment, 10 bytes + x bytes
 R: Reserved, 4 bytes

Process data sent from the fieldbus coupler to the higher-level controller:

Physical structure



DI: Digital input, 8 bits
 DO: Digital output, 8 bits
 AI: Analog input, 8 bytes
 AO: Analog output, 8 bytes
 F_Var: Fieldbus output variables, 8 bytes
 Error: 3 bytes
 Status: 1 byte
 Diagnostics: 10 bytes + x bytes
 Error group: 1 byte
 Error code: 1 byte
 Error argument: 1 byte

12.4 Diagnostics

Certain diagnostic information is made available to you in the network. This can include, for example, error messages and warnings from the fieldbus coupler, the IEC 61131 application or the connected I/O modules. You have several options for receiving diagnostic information from the fieldbus coupler:

- Diagnostics integrated in the input and output data,
- Diagnostics via explicit messages,
- Diagnostics mixed via input and output data and explicit messages

The diagnostic information of all connected I/O modules and of the fieldbus coupler is stored in the latter for collection. When storing diagnostic information, bit 3 is also placed in the status byte. If more diagnostic information is reported to the fieldbus coupler than is collected, an overflow of the internal memory can occur. If this condition arises, bit 2 is placed in the status byte, through which a loss of diagnostic information is reported. From this point on, the respective current diagnostic information is stored and the oldest is discarded. More information on status bytes can be found in the next section and under attribute 5 of the configuration class (18.1.1) of the fieldbus coupler.

The following sections describe how to access diagnostic information via the Ethernet/IP protocol. As with the process data, the data here that is larger than one byte is displayed in Intel format ("Little Endian").

Class:

0x0009 designates the "Digital Output Point" class as an error class assigned to the digital output modules.

Instance:

The instance 0x0005 identifies channel 5 of the digital output module as the source of the error.

Attribute:

The attribute 0x0081 (129) indicates an "Open Load" error.

The description of the diagnostic source on the basis of class, instance and attribute (CIA) can be found in the manuals of the respective I/O modules (appendix).

Number of valid data bytes:

The length of the diagnostic data is identified by the I/O module that can deliver the greatest number of diagnostic bytes. If only a portion of the diagnostic data is used in a current diagnostic message, this portion is determined by the "Number of valid data bytes". Beginning with the next byte, only the number of bytes specified here is valid.

In the example, the value 0x01 indicates that only the next byte in the diagnostic data contains a valid value. The subsequent 3 bytes are reserved and their content is irrelevant.

Diagnostic data:

The first byte of diagnostic data contains a 1 and thus indicates an incoming diagnosis on channel 3 of the digital output module, such as "Open Load". The value 0 indicates an outgoing diagnosis.

Some diagnostic messages require a confirmation. By means of the assembly instance 113, the diagnostic confirmations are transmitted from the higher-level controller to the fieldbus coupler. The data contents are shown in the following table:

Data from fieldbus coupler to higher-level controller				
Process image	Error (3 bytes)	Status byte	Diagnostic message (10 + x bytes)	Fieldbus output variables
Data from higher-level controller to fieldbus coupler				
Process image	Reserved (4 bytes)		Diagnostic confirmation (10 + x bytes)	Fieldbus output variables

By reserving 4 bytes, the beginning of the diagnostic message and diagnostic confirmation is identical as it relates to the process image (offset of 4 bytes).

Note



If several diagnostic messages are pending simultaneously, they are transmitted back-to-back. Therefore, the individual diagnostic messages are only available in a cycle, i.e. in an Ethernet telegram. If available, the next diagnostic message would then be transmitted in the next Ethernet telegram.

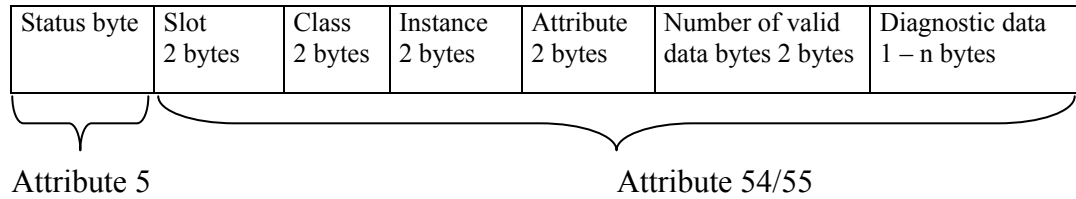
Information



To confirm the diagnosis, copy the data from the diagnostic message into the diagnostic confirmation.

12.4.2 Diagnostics via Explicit Messages

Using explicit messages, the complete diagnosis can also be read out via attributes. The individual data contents of the diagnostic message or confirmation can be read out from the attributes of class 64_{hex}, instance 1.



Attribute 5: This attribute contains the status byte.

Attribute 54: This contains the diagnostic message which is composed of slot, class, instance, attribute, number of valid data bytes and diagnostic data.

Attribute 55: This contains the diagnostic confirmation which is composed of slot, class, instance, attribute, number of valid data bytes and diagnostic data.

The descriptions of individual contents, such as slot, etc., can be found in Section 12.4.1.

The status byte can be read out via attribute 5 by bit 3 indicating whether or not a diagnosis is pending. This bit remains set as long as diagnostic messages are pending. If the diagnosis is only processed via explicit messages, attribute 5 should be read out in certain time intervals in order to detect a diagnostic message. If a diagnostic message is pending, the diagnostic source can be read out via attribute 54 of this class. If a confirmation of the diagnosis is necessary, the user can employ attribute 55 for this purpose.

The data structure of attributes 54 and 55 corresponds to the structure as transmitted in the input and output data (see figure above). Please ensure that the data is transmitted and confirmed in "Little Endian" format. An absence of an available diagnosis is indicated by resetting bit 3 from attribute 5. In addition, the entire data content of attribute 54 is set to zero.

Note



A diagnostic message can only be read out via attribute 54. If another read action is implemented, either the next pending diagnostic message is delivered or none at all (data content of attribute 54 is zero).

Information



To confirm the diagnosis, copy the data from the diagnostic message into the diagnostic confirmation.

12.4.3 Diagnostics Mixed via Input and Output Data and Explicit Messages

There are several options for receiving and confirming the diagnosis mixed via input and output data and explicit messages. The assembly instances 104, 105 and 106 offer these alternatives:

- **Instance 104**
For analog and digital input data, including status byte.
- **Instance 105**
For digital input data, including status byte.
- **Instance 106**
For analog input data, including status byte.

In these assembly instances, the status byte is integrated and gives information about a pending diagnosis (bit 3 set). The actual diagnosis can then be read out by means of the explicit messages from class 100, instance 1, attribute 54. With attribute 55 of the same class, the diagnosis is confirmed. For more information, see Section 12.4.2.

Other possibilities for combinations of input and output data and explicit messages are also available to link the diagnostic information to the overall concept.

12.5 Object Model

For network communication, EtherNet/IP utilizes an object model in which all functions and data of the fieldbus coupler 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 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:**
A specific and physical occurrence of an object. For example, New Zealand is an instance of the "land" object class. The terms "object," "instance" and "object instance" all refer to a specific instance.
- **Attribute:**
The description of an externally visible characteristic or of an object's function. Attributes typically deliver status information or regulate the function of an object. For example, the ASCII name of an object and the repetition frequency of a periodic object.
- **Service:**
A function supported by an object and/or an object class. CIP defines a group of common services that are applied to the attributes.
- **Behavior:**
Specification of 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.

12.6 CIP Classes

The CIP class are included in the CIP specification of ODVA. They describe the properties (band 1) of Ethernet and CAN independent of their physical interface. The physical interface is described in a separate specification. For EtherNet/IP, this is band 2, which describes the adaption of EtherNet/IP to CIP.

For this purpose, WAGO uses classes 1, 2, 4 – 6 and also F4_{hex}, which are described in band 1 ("Common Industrial Protocol"). Classes F5_{hex} and F6_{hex} are supported from band 2 (EtherNet/IP Adaption of CIP). The classes in the following table are described in more detail in subsequent sections. The description of WAGO-specific classes begins in Section 12.7.

Table 22: CIP classes

Class	Name	Information
01 _{hex}	Identity object	See Section 12.6.1
02 _{hex}	Message router object	See Section 12.6.2
04 _{hex}	Assembly object	See Section 12.6.3
05 _{hex}	Connection object	See Section 12.6.4
06 _{hex}	Connection manager object	See Section 12.6.5
F4 _{hex}	Port class object	See Section 12.6.6
F5 _{hex}	TCP/IP interface object	See Section 12.6.7
F6 _{hex}	Ethernet link object	See Section 12.6.8

Table 23: Explanation of table headers

Column heading	Description
Attribute ID	Integer value which is assigned to the respective attribute
Access	Set: The attribute can be accessed using the Set_Attribute service (writing/modifying the attribute value). Important: If an attribute supports the Set_Attribute service, it can also be accessed with the Get_Attribute service. Get: The attribute can be accessed using the Get_Attribute services (reading the attribute value). Get_Attribute_All: Delivers the content of all attributes. Set_Attribute_Single: Modifies an attribute value. Reset: Performs a restart. 0: Emulates a restart. 1: Emulates a restart and restores the default settings.

Table 23: Explanation of table headers

Column heading	Description
NV	NV (non-volatile): The attribute is permanently saved in the fieldbus coupler. V (volatile): The attribute is not permanently saved in the fieldbus coupler.
Name	Identification of the attribute.
Data type	Identification of the CIP data type of the attribute.
Description	Brief description of the attribute.
Default value	Factory setting

12.6.1 Identity Object (01_{hex})

The identity class provides general information about the fieldbus coupler that clearly identifies it.

Class attributes

Table 24: "Identity object" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of 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 (only applies to Get_Attribute_All service)	0 (0x0000)
4	Get	Max ID Number of Instance Attribute	UINT	Maximum number of instance attributes (only applies to Get_Attribute_All service)	0 (0x0000)

Instance attribute 1

Table 25: "Identity object" class: instance attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Vendor ID	UINT	Manufacturer identification	40 (0x0028)
2	Get	Device Type	UINT	General designation of fieldbus coupler type	12 (0x000C)
3	Get	Product Code	UINT	Identification of fieldbus coupler	2301 (0x08FD)
4	Get	Revision	STRUCT of:	Version of identity object	Firmware dependent
		Major Revision	USINT		
		Minor Revision	USINT		

Table 25: "Identity object" class: instance attribute

Attribute ID	Access	Name	Data type	Description	Default value
5	Get	Status	WORD	See *	Dependent on current status
6	Get	Serial Number	UDINT	Serial number	-
7	Get	Product Name	SHORT_String	Product name	2301: WAGO PFC ETHERNET 8DI 24 V DC

Overview of services

Table 26: "Identity object" class: services

Service code	Service available		Service name	Description
	Class	Instance		
01 hex	Yes	Yes	Get_Attribute_All	Delivers content of all attributes
05 hex	No	Yes	Reset	Carries out the reset service. Service parameter: 0: Emulates a restart 1: Emulates a restart and restores the default settings
0E hex	No	Yes	Get_Attribute_Single	Delivers content of respective attribute

* **Bit 0:** Assignment to a master; **Bit 1=0** (reserved); **Bit 2:** Configured: (= 0: configuration is unchanged; =1: configuration differs from manufacturer's parameters); **Bit 3 = 0** (reserved); **Bit 4-7:** Extended device status: (= 0010: at least one faulty I/O connection, = 0011: no I/O connection established);

Bit 8-11: unused

Bit 12-15 = 0 (reserved)

12.6.2 Message Router Object (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.

Class attributes

Table 27: "Message router object" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of 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



The class attributes are only accessible with the Get_Attribute_All service.

Instance attribute 1

Table 28: "Message router object" class: instance attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	ObjectList	STRUCT of:	-	
		Number	UINT	Number of implemented classes	39 (0x0027)
		Classes	UINT	Implemented classes	0x0001, 0x0002, 0x0004, 0x0006, 0x00F4, 0x00F5, 0x00F6, 0x0064 – 0x0074, 0x0082, 0x00A0, 0x00A1, 0x00A2, 0x00A6, 0x00A7, 0x00AA, 0x00AB, 0x00A3, 0x00A4, 0x00A5, 0x00A8, 0x00A9, 0x00AC, 0x00AD
2	Get	NumberAvailable	UINT	Maximum number of differing connections	128 (0x0080)

Overview of services

Table 29: "Message router object" class: services

Service code	Service available		Service name	Description
	Class	Instance		
01 hex	Yes	No	Get_Attribute_All	Delivers content of all attributes
0E hex	No	Yes	Get_Attribute_Single	Delivers content of respective attribute

12.6.3 Assembly Object (04_{hex})

By means of the assembly classes, 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 instances with their contents and arrangements.

Class attributes

Table 30 "Assembly object" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	2 (0x0002)
2	Get	Max Instance	UINT	Highest instance	113 (0x0071)

Instance attribute 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 31: "Assembly object" class: instance attribute 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 fieldbus input variables, are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 102 (66 hex)

This assembly instance contains digital output data and fieldbus input variables only.

Table 32: "Assembly object" class: instance attribute 102 (66 hex)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get/Set	Data	ARRAY of BYTE	Only digital output data and fieldbus input variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 103 (67 hex)

This assembly instance contains analog output data and fieldbus input variables only.

Table 33: "Assembly object" class: instance attribute 103 (67 hex)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get/Set	Data	ARRAY of BYTE	Only analog output data and fieldbus input variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 104 (68 hex)

This assembly instance contains analog and digital input data, the status and fieldbus output variables only.

Table 34: "Assembly object" class: instance attribute 104 (68 hex)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Only analog and digital input data, the status and fieldbus output variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 105 (69 hex)

This assembly instance contains digital input data, the status and fieldbus output variables only.

Table 35: "Assembly object" class: instance attribute 105 (69 hex)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Only digital input data, the status and fieldbus output variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 106 (6A_{hex})

This assembly instance contains analog input data, the status and fieldbus output variables only.

Table 36: "Assembly object" class: instance attribute 106 (6A_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Only analog input data, the status and fieldbus output variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 107 (6B_{hex})

This assembly instance contains analog and digital input data and fieldbus output variables.

Table 37: "Assembly object" class: instance attribute 107 (6B_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Only analog and digital input data and fieldbus output variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 108 (6C_{hex})

This assembly instance contains digital input data and fieldbus output variables.

Table 38: "Assembly object" class: instance attribute 108 (6C_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Only digital input data and fieldbus output variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 109 (6D_{hex})

This assembly instance contains analog input data and fieldbus output variables.

Table 39: "Assembly object" class: instance attribute 109 (6D_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Only analog input data and fieldbus output variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 110 (6E_{hex})

This assembly instance contains fieldbus output variables.

Table 40: "Assembly object" class: instance attribute 110 (6E_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Only fieldbus output variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 111 (6F_{hex})

This assembly instance contains fieldbus input variables only.

Table 41: "Assembly object" class: instance attribute 111 (6F_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get/Set	Data	ARRAY of BYTE	Only fieldbus input variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 112 (70_{hex})

This assembly instance contains analog and digital input data, current errors, the status, diagnosis and fieldbus input variables.

Table 42: "Assembly object" class: instance attribute 112 (70_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Only analog and digital input data, current errors, the status, diagnosis and fieldbus input variables are contained in the process image	-
4	Get	Data Size	ARRAY of BYTE	Number of bytes in process image	-

Instance attribute 113 (71_{hex})

This assembly instance contains analog and digital output data, 4 reserved bytes*, the status and the diagnostic confirmation.

Table 43: "Assembly object" class: instance attribute 113 (71_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get/Set	Data	ARRAY of BYTE	Only analog and digital output data, 4 reserved bytes*, the status, the diagnostic confirmation and the fieldbus input variables are contained in the process image	-
4		Data Size	ARRAY of BYTE	Number of bytes in process image	

Overview of services

Table 44: "Assembly object" class; services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

The software inspects the writing of attribute 3 of assembly instances 101 through 103 and 113. 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.

* These reserved bytes merely assist in providing the identical data width of the status and diagnostic information as the assembly instance 112: 1 byte status + 3 byte error. Thus, you can expect the same offset between the analog or digital data and the fieldbus variables.

Instance attribute 198 (C6_{hex}) "input only"

This instance assists in setting up the connection when no outputs are to be attached or when inputs are requested that are already being used in an exclusive owner connection. The data length of this instance is always zero.

This instance can only be used in the "consumed path" (from the viewpoint of the slave).

Instance attribute 199 (C7_{hex}) "listen only"

With this instance, a connection can be set up that attaches to an existing exclusive owner connection. Here the new connection has the same transmission parameters as the exclusive owner connection. If the exclusive owner connection is terminated, this connection is likewise automatically terminated. The data length of this instance is always zero.

This instance can only be used in the "consumed path" (from the viewpoint of the slave).

12.6.4 Connection Object (05_{hex})

Because the connections are established and terminated via the connection manager, the class and instance attributes of this class are not visible.

12.6.5 Connection Manager Object (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.

12.6.6 Port Class Object (F4_{hex})

The "port class object" specifies the existing CIP ports on the fieldbus coupler. There is one instance for each CIP port.

Class attribute

Table 45: "Port class object" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	1 (0x0001)
3	Get	Num Instances	UINT	Number of current ports	1 (0x0001)
8	Get	Entry Port	UINT	Instance of port object from which the request arrived	1 (0x0001)
9	Get	All Ports	Array of Struct UINT UINT	Array of instance attributes 1 and 2 of all instances	0 (0x0000) 0 (0x0000) 4 (0x0004) 2 (0x0002)

Instance attribute 1

Table 46: "Port class object" class: instance attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Port Type	UINT	-	4 (0x0004)
2	Get	Port Number	UINT	CIP port number	2 (0x0002) (EtherNet/IP)
3	Get	Port Object	UINT	Number of 16-bit words in the subsequent path	2 (0x0002)
			Padded EPATH	Object that manages this port	0x20 0xF5 0x24 0x01 (corresponds to TCP/IP interface object)
4	Get	Port Name	Short String	Port name	“ ”
7	Get	Node Address	Padded EPATH	Port segment (IP address)	Depends on IP address

Overview of services

Table 47: "Port class object" class: services

Service code	Service available		Service name	Description
	Class	Instance		
01 hex	Yes	Yes	Get_Attribute_All	Delivers content of all attributes
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

12.6.7 TCP/IP Interface Object (F5_{hex})

The "TCP/IP interface object" provides for the configuration of the TCP/IP network interface of a fieldbus coupler. Examples of configurable objects include the IP address, the network mask and the gateway address of the fieldbus coupler.

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 interface or a serial interface for protocols such as PPP. 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.

Class attribute

Table 48: "TCP/IP interface object" class: class attributes

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	1 (0x0001)
3	Get	Num Instance	UINT	Number of currently instanced connections	1 (0x0001)

Instance attribute 1

Table 49: "TCP/IP interface object" class: instance attributes

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Status	DWORD	Interface status	-
2	Get	V	Configuration Capability	DWORD	Interface flags for possible configuration types	23 (0x00000017)
3	Set	NV	Configuration Control	DWORD	Determines how the fieldbus coupler arrives at its TCP/IP configuration after the first restart	17 (0x00000011)

Table 49: "TCP/IP interface object" class: instance attributes

Attribute ID	Access	NV	Name	Data type	Description	Default value
4	Get	V	Physical Link Object	STRUCT of	-	
			Path	UINT	Number of 16-bit words in the subsequent path	2 (0x0002)
				Padded EPATH	Logical path that points to the physical link object	0x20 0xF6 0x24 0x03 (corresponds to the Ethernet link object)
5	Set	NV	Interface Configuration	STRUCT of	-	
			IP Address	UDINT	IP address	0
			Network Mask	UDINT	Network mask	255 (0xFF) 255 (0xFF) 0 (0x00) 0 (0x00)
			Gateway Address	UDINT	IP address of the standard gateway	0
			Name Server	UDINT	IP address of the primary name server	0
			Name Server 2	UDINT	IP address of the secondary name server	0
			Domain Name	STRING	Standard domain name	„“
6	Set	NV	Host Name	STRING	Device name	MAC ID

Overview of services

Table 50: "TCP/IP interface object" class: services

Service code	Service available		Service name	Description
	Class	Instance		
01 hex	Yes	Yes	Get_Attribute_All	Delivers content of all attributes
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.6.8 Ethernet Link Object (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.

Class attribute

Table 51: "Ethernet link object" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	3 (0x0003)
2	Get	Max Instance	UDINT	Maximum number of instances	3 (0x00000003)
3	Get	Num Instances	UDINT	Number of currently instanced connections	3 (0x00000003)

Instance attribute 1

Table 52: "Ethernet link object" class: instance attribute 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Interface Speed	UDINT	Transfer rate	10 (0x0A) oder 100 (0x64)
2	Get	Interface Flags	DWORD	Interface configuration information/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 Adresse	MAC ID of fieldbus coupler
6	Set	Interface Control	STRUCT of:	Configuration of physical interface	
		Control Bits	WORD	Interface control bits Bit 0: Automatic detection Bit 1: Default duplex mode Bit 2 – 15: Reserved	0x0001
		Forced Interface Speed	UINT	Preset interface speed	0 (0x0000)

Table 52: "Ethernet link object" class: instance attribute 1

Attribute ID	Access	Name	Data type	Description	Default value
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: Glass fiber (e.g. 100Base-FX). Values 4 – 256: Reserved	2 (0x02)
8	Get	Interface State	USINT	Interface status: Value 0: Unknown Value 1: Interface active and ready to send/receive. Value 2: Interface deactivated. Value 3: Interface is testing. Values 4 – 256: Reserved	-
9	Set	Admin State	USINT	Administration status: Value 0: Reserved Value 1: Activate interface. Value 2: Deactivate interface. If this is the sole CIP interface, a request to deactivate is acknowledged with an error (error code 0x09) Values 3 – 256: Reserved	1 (0x01)
10	Get	Interface Label	SHORT_STRING	Readable identification	"Port 1"

Instance attribute 2

Table 53: "Ethernet link object" class: instance attribute 2

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Interface Speed	UDINT	Transfer rate	10 (0x0A) oder 100 (0x64)
2	Get	Interface Flags	DWORD	Interface configuration information/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 Adresse	MAC ID of fieldbus coupler
6	Set	Interface Control	STRUCT of:	Configuration of physical interface	-
		Control Bits	WORD	Interface control bits Bit 0: Automatic detection Bit 1: Default duplex mode Bit 2 – 15: Reserved	1 (0x0001)
		Forced Interface Speed	UINT	Preset interface speed	0 (0x0000)
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: Glass fiber (e.g. 100Base-FX). Values 4 – 256: Reserved	2 (0x02)
8	Get	Interface State	USINT	Interface status: Value 0: Unknown Value 1: Interface active and ready to send/receive. Value 2: Interface deactivated. Value 3: Interface is testing. Values 4 – 256: Reserved	-

Table 53: "Ethernet link object" class: instance attribute 2

Attribute ID	Access	Name	Data type	Description	Default value
9	Set	Admin State	USINT	Administration status: Value 0: Reserved Value 1: Activate interface. Value 2: Deactivate interface. If this is the sole CIP interface, a request to deactivate is acknowledged with an error (error code 0x09) Values 3 – 256: Reserved	1 (0x01)
10	Get	Interface Label	SHORT_STRING	Readable identification	"Port 2"

Instance attribute 3

Table 54: "Ethernet link object" class: instance attribute 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 information/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	-
3	Get	Physical Address	ARRAY of 6 UINTs	MAC Adresse	MAC ID of fieldbus coupler
6	Get	Interface Control	STRUCT of:	Configuration of physical interface	
		Control Bits	WORD	Interface control bits Bit 0: Automatic detection Bit 1: Default duplex mode Bit 2 – 15: Reserved	0x0002
		Forced Interface Speed	UINT	Preset interface speed	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: Glass fiber (e.g. 100Base-FX). Values 4 – 256: Reserved	1 (0x01)

Table 54: "Ethernet link object" class: instance attribute 3

Attribute ID	Access	Name	Data type	Description	Default value
8	Get	Interface State	USINT	Interface status: Value 0: Unknown Value 1: Interface active and ready to send/receive. Value 2: Interface deactivated. Value 3: Interface is testing. Values 4 – 256: Reserved	-
9	Get	Admin State	USINT	Administration status: Value 0: Reserved Value 1: Activate interface. Value 2: Deactivate interface. If this is the sole CIP interface, a request to deactivate is acknowledged with an error (error code 0x09) Values 3 – 256: Reserved	1 (0x01)
10	Get	Interface Label	SHORT_STRING	Readable identification	"Internal port 3"

Overview of services

Table 55: "Ethernet link object" class: services

Service code	Service available		Service name	Description
	Class	Instance		
01 hex	Yes	Yes	Get_Attribute_All	Delivers content of all attributes
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.7 WAGO-Specific Classes

This manufacturer-specific addition contains information on the 767 components that is not contained in the CIP classes. The WAGO-specific classes are listed in the appendix.

Table 56: WAGO-specific classes

Name	Class	Information
Coupler Configuration Object	64 _{hex}	See Section 18.1.1
Node Information Class	82 _{hex}	See Section 18.1.6
Discrete Input Point	65 _{hex}	See Section 18.1.2.1
Discrete Input Point Extended 1	69 _{hex}	See Section 18.1.2.2
Discrete Input Point Extended 2	6D _{hex}	See Section 18.1.2.3
Discrete Input Point Extended 3	71 _{hex}	See Section 18.1.2.4
Discrete Output Point	66 _{hex}	See Section 18.1.3.1
Discrete Output Point Extended 1	6A _{hex}	See Section 18.1.3.2
Discrete Output Point Extended 2	6E _{hex}	See Section 18.1.3.3
Discrete Output Point Extended 3	72 _{hex}	See Section 18.1.3.4
Analog Input Point	67 _{hex}	See Section 18.1.4.1
Analog Input Point Extended 1	6B _{hex}	See Section 18.1.4.2
Analog Input Point Extended 2	6F _{hex}	See Section 18.1.4.3
Analog Input Point Extended 3	73 _{hex}	See Section 18.1.4.4
Analog Output Point	68 _{hex}	See Section 18.1.5.1
Analog Output Point Extended 1	6C _{hex}	See Section 18.1.5.2
Analog Output Point Extended 2	70 _{hex}	See Section 18.1.5.3
Analog Output Point Extended 3	74 _{hex}	See Section 18.1.5.4
Input Fieldbus Variable USINT	A0 _{hex}	See Section 18.1.9.1
Input Fieldbus Variable USINT Extended 1	A1 _{hex}	See Section 18.1.9.2
Input Fieldbus Variable USINT Extended 2	A2 _{hex}	See Section 18.1.9.3
Input Fieldbus Variable UINT	6A _{hex}	See Section 18.1.9.4
Input Fieldbus Variable UINT Extended 1	A7 _{hex}	See Section 18.1.9.5
Input Fieldbus Variable UDINT	AA _{hex}	See Section 18.1.9.6
Input Fieldbus Variable UDINT Offset UINT	AB _{hex}	See Section 18.1.9.7
Output Fieldbus Variable USINT	A3 _{hex}	See Section 18.1.10.1
Output Fieldbus Variable USINT Extended 1	A4 _{hex}	See Section 18.1.10.2

Table 56: WAGO-specific classes

Name	Class	Information
Output Fieldbus Variable USINT Extended 2	A5 _{hex}	See Section 18.1.10.3
Output Fieldbus Variable UINT	A8 _{hex}	See Section 18.1.10.4
Output Fieldbus Variable UINT Extended 1	A9 _{hex}	See Section 18.1.10.5
Output Fieldbus Variable UDINT	AC _{hex}	See Section 18.1.10.6
Output Fieldbus Variable UDINT Offset UINT	AD _{hex}	See Section 18.1.10.7

13 MODBUS

The 767 Series modular concept makes it possible to attach up to 64 external I/O modules to the fieldbus coupler. However, the variable node design as well as the large number of differing I/O modules make it impossible to statically assign input and output data to fixed MODBUS addresses. The only exceptions are the "digital" MODBUS services. For these, the MODBUS address is identical to the channel number, i.e., the 47th digital input can always be found at the MODBUS address "46."

By adding or removing I/O modules, the structure of the process images is changed which also changes the MODBUS addresses of the individual I/O module channels.

The MODBUS communication is performed via service calls. To do this, the MODBUS master (client) sends a request telegram to port 502 of the MODBUS slave (server). The MODBUS slave returns the result of the service call in a response telegram to the MODBUS master.

The most important elements of a MODBUS telegram are:

Term	Description
FunctionCode (FC)	Service identifier: Read or write operation to bits or words
Address	Operation start address
Count	Number of bits or words depending on the service
[Data]	Process data

The service identifier or "FunctionCode" (FC) first determine whether a read or write operation is involved. It also determines the basic data type to which the operation is to be applied. Therefore, the meaning of the parameters "Address" and "Count" is also dependent on the function code. Thus "address :=3" can stand for a bit or a word in the input or output process image.

The MODBUS protocol is largely based on the following basic data types:

Data Type	Length	Description
Discrete Inputs	1 bit	Digital inputs
Coils	1 bit	Digital outputs
Input Register	16 bits	Analog input data
Holding Register	16 bits	Analog output data

One or more "FunctionCodes" are defined for each basic data type.

Although digital and analog process data from the fieldbus coupler and I/O modules is combined in a process image, the first digital output or input is always reached with the "digital" MODBUS services at address 0. You get to the first analog output or input via the word services.

Note



The fieldbus coupler supports up to 15 socket connections simultaneously via MODBUS/TCP. For MODBUS/UDP, there are up to 5 telegrams that can be stored temporarily.

13.1 Process Image

After starting up the fieldbus coupler, it automatically identifies all connected I/O modules. The fieldbus coupler uses this to create a local process image, divided into an input and output data range. The data from the analog I/O modules and subsequently the data from the digital I/O modules are initially stored in these areas.

The size of the process image can be identified from the connected 767 components.

13.2 Process Data Exchange

Data exchange between the MODBUS/TCP master and the I/O modules is achieved using the MODBUS functions implemented in the fieldbus coupler via bit-by-bit or word-by-word reading and writing routines. There are four different types of process data in the fieldbus coupler:

- Input words
- Output words
- Input bits
- Output bits

13.3 Accessing the Process Image via MODBUS Functions

The following table describes the access types, which can be used to access logical address ranges.

Table 57: MODBUS function codes (FC)

FC	Name	Description
FC1	Read coils	Several digital output values are read out
FC2	Read inputs discrete	Read several digital input and output values
FC3	Read holding registers	Read several analog input and output values
FC4	Read input registers	Read several analog input and output values
FC5	Write coil	A single digital output value is written
FC6	Write single register	A single analog output value is written
FC15	Force multiple coils	Several digital output values are written
FC16	Write multiple registers	Several analog output values are written
FC22	Write register screen	Write individual bits to a register using an AND or OR screen
FC23	Read/write multiple registers	Write and read operation on analog input and output values

Note



Note the number system when addressing!

The examples listed use the hexadecimal system (e.g., 0x000) as the number format. Addressing starts with 0.

Depending on the software controller, the format and start of addressing can vary. In this case, all addresses are to be converted accordingly.

13.3.1 Register Services

With the register services, you can identify or modify the status of analog input and output modules using the function codes FC3, FC4 and FC6, FC16, FC22 and FC23.

Tabelle 58: Read analog input terminals using FC3, FC4, FC23

MODBUS address		Description
HEX	DEC	
0x0000 – 0x00FF	0 – 255	Read analog or digital input values (part 1). Physical address space of the input data of 256 words.
0x0100 – 0x01FF	256 – 511	Read PLC output variables (PFC-OUT). Using a template, you can access PLC output variables via %QW256 – %QW511 addresses in CoDeSys 3.
0x0200 – 0x02FF	512 – 767	Read back of analog or digital output values (part 1). Physical size of the address space: 256 words.
0x0300 – 0x03FF	768 – 1023	Read PLC input variables (PFC-IN).
0x0400 – 0x0FFF	1024 – 4095	Invalid range ("Illegal data address")
0x1000 – 0x2FFF	4096 – 12287	Configuration register. See Section 13.3.3.
0x3000 – 0x3FFF	12288 – 16383	Read flag variables. Physical size of the address space: 8 kB.
0x4000 – 0x5FFF	16384 – 24575	Invalid range ("Illegal data address")
0x6000 – 0x62FB	24576 – 25339	Read the input data (part 2). Additional address space of 764 words.
0x62FC – 0x6FFF	25340 – 28671	Invalid range ("Illegal data address")
0x7000 – 0x72FB	28672 – 29435	Read output data (part 2). Additional address space of 764 words.
0x72FC – 0x7FFF	29436 – 32767	Invalid range ("Illegal data address")
0x8000 – 0x8FFF	32768 – 36863	Read retain variables that were written by MODBUS. Additional address space of 8kB retain memory. CoDeSys (PLC) can only read this space!
0x9000 – 0xFFFF	36864 – 65535	Invalid range ("Illegal data address")

Tabelle 59: Schreiben von Analogausgangsklemmen mittels FC6, FC16, FC22, FC23

MODBUS address		Description
HEX	DEC	
0x0000 – 0x00FF	0 – 255	Write analog or digital output values (part 1). Physical address space of the output data of 256 words.
0x0100 – 0x01FF	256 – 511	Write PLC input variables (PFC-IN). Using a template, you can access PLC input variables via %IW256 – %IW511 addresses in CoDeSys 3.
0x0200 – 0x02FF	512 – 767	Write analog or digital output values (part 1). Physical size of the address space: 256 words. Note: Here you write the same output values of the address range 0x0000 – 0x00FF.
0x0300 – 0x03FF	768 – 1023	Write PLC input variables (PFC-IN). Using a template, you can access PLC input variables via %IW256 – %IW511 addresses in CoDeSys 3.
0x0400 – 0x0FFF	1024 – 4095	Invalid range ("Illegal data address")
0x1000 – 0x2FFF	4096 – 12287	Configuration register. See Section 13.3.3.
0x3000 – 0x3FFF	12288 – 16383	Write retain variables (NOVRAM). Additional address space of 8kB retain memory. CoDeSys (PLC) can only read this space!
0x4000 – 0x5FFF	16384 – 24575	Invalid range ("Illegal data address")
0x6000 – 0x62FB	24576 – 25339	Write the output data (part 2). Additional address space of 764 words.
0x62FC – 0x6FFF	25340 – 28671	Invalid range ("Illegal data address")
0x7000 – 0x72FB	(28672 – 29435)	Write the output data (part 2). Additional address space of 764 words. Note: Here you write the same output values of the address range 0x6000 – 0x62FC.
0x72FC – 0x7FFF	29436 – 32767	Invalid range ("Illegal data address")

13.3.2 Bit Services

With the digital bit services, you can identify or modify the status of digital input and output modules using the function codes FC1, FC2 and FC5, FC15.

Tabelle 60: Read digital input terminals using FC1, FC2

MODBUS address		Description
HEX	DEC	
0x0000 – 0x01FF	0 – 511	Read the first 512 input values
0x0200 – 0x03FF	512 – 1023	Read the first 512 output values
0x0400 – 0x0FFF	1024 – 4095	Invalid range ("Illegal data address")
0x1000 – 0x1FFF	4096 – 8191	Read PLC output variables (PFC-OUT). Using a template, you can access PLC output variables via %QX256.0 – %QX511.15 addresses in CoDeSys 3.
0x2000 – 0x2FFF	8192 – 12287	Read back PLC input variables. Using a template, you can access PLC input variables via %IX256.0 – %IX511.15 addresses in CoDeSys 3.
0x3000 – 0x7FFF	12288– 32767	Read volatile flag variables. Physical size of the address space: 8 kB.
0x8000 – 0x85F7	32768– 34295	Read the input values in the range of 513 to 2039
0x85F8 – 0x8FFF	34296– 36863	Invalid range ("Illegal data address")
0x9000 – 0x95F7	36864– 38391	Read the output values in the range of 513 to 2039
0xA000 – 0xEFFF	40960– 61439	Read retain variables that were written by MODBUS. Additional address space of 8kB retain memory. CoDeSys (PLC) can only read this space!
0xF000 – 0xFFFF	61440– 65535	Invalid range ("Illegal data address")

Tabelle 61: Write digital output terminal using FC5, FC15

MODBUS address		Description
HEX	DEC	
0x0000 – 0x01FF	0 – 511	Write the first 512 output values.
0x0200 – 0x03FF	512 – 1023	Write the first 512 output values.
0x0400 – 0x0FFF	1024 – 4095	Invalid range ("Illegal data address")
0x1000 – 0x1FFF	4096 – 8191	Write PLC input variables (PFC-IN). Using a template, you can access PLC input variables via %IX256.0 – %IX511.15 addresses in CoDeSys 3.
0x2000 – 0x2FFF	8192 – 12287	Write PLC input variables (PFC-IN). Using a template, you can access PLC input variables via %IX256.0 – %IX511.15 addresses in CoDeSys 3. Note: Write the same output values for addresses 0x1000 – 0x1FFF
0x3000 – 0x7FFF	12288 – 32767	Write retain variables. Additional address space of 8kB retain memory. CoDeSys (PLC) can only read this space!
0x8000 – 0x85F7	32768 – 34295	Write the output values in the range of 513 to 2039.
0x85F8 – 0x8FFF	34296 – 36863	Invalid range ("Illegal data address")
0x9000 – 0x95F7	36864 – 38391	Write the output values in the range of 513 to 2039.
0x95F8 – 0xFFFF	38392 – 65535	Invalid range ("Illegal data address")

13.3.3 Configuration Register

By using the configuration register, you can configure the fieldbus coupler and read information using it. Access register using MODBUS functions FC4 "Read Input Registers" and FC6 "Write Single Register".

Table 62: Configuration register for MODBUS V1

Address	Access	Length (word)	Description
0x1000	R/W	1	Read/write watchdog time
0x1001	R/W	1	Watchdog coding screen 1-16
0x1002	R/W	1	Watchdog coding screen 17-32
0x1003	R/W	1	Watchdog trigger
0x1004	R	1	Minimum trigger time
0x1005	R/W	1	Stop watchdog (write sequence 0xAAAA, 0x5555)
0x1006	R	1	Watchdog status
0x1007	R/W	1	Watchdog restart (write sequence 0x1)
0x1008	RW	1	Stop watchdog (write sequence 0x55AA or 0xAA55)
0x1009	R/W	1	MODBUS and http connection stop at a watchdog timeout
0x100A	R/W	1	Watchdog configuration
0x100B	W	1	Save watchdog parameters
0x1020	R	1 – 2	Display error group via LED
0x1021	R	1	Display error argument and error code via LED
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 input data in the process image (in bits)
0x1025	R	1	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
0x1030	R/W	1	Configuration MODBUS/TCP timeout
0x1031	R	1	Read the MAC ID of the fieldbus coupler
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 index
0x2011	R	1	Series designation (767)
0x2012	R	1	Fieldbus type (2301)
0x2013	R	1	Firmware versions (major revision)
0x2014	R	1	Firmware versions (minor revision)
0x2020	R	16	Fieldbus coupler short description
0x2021	R	8	Firmware compile time
0x2022	R	8	Firmware compile date
0x2023	R	32	Firmware loader version

Table 62: Configuration register for MODBUS V1

Address	Access	Length (word)	Description
0x2030	R	65	Description of the connected I/O modules (0 – 64)
0x2040	W	1	Software reset (write sequence 0x55AA or 0xAA55)
0x2041	W	1	Formate the user drive
0x2042	W	1	Extract the HTML pages from the firmware
0x2043	W	1	Default settings
0x2099	R/W	1	MODBUS version change (compatibility mode for Release 1)

13.3.4 Watchdog Behavior

A timing function (timeout) is periodically initiated in the fieldbus coupler by the higher-level controller for monitoring the TCP connection. Because this time is always restarted before communication takes place, it cannot reach its upper value in the case of error-free communication. If this time has run out, a fieldbus failure exists. In this case, the fieldbus coupler answers all subsequent MODBUS "TCP/IP requests" with the exception code 0x0004 (slave device failure). Separate registers exist in the fieldbus coupler for controlling the watchdog and requesting its status through the higher-level controller.

The watchdog is not yet activated when the power supply is switched on. The timeout value must first be determined (register 0x1000). The watchdog can be activated by writing a value other than 0 in the register (0x1001).

13.3.4.1 Watchdog Register

Table 63: Register address 0x1000

Register address 0x1000 (4096 _{dec})	
Value	Watchdog time
Access	Read/write
Default	0x0000
Description	The register saves the value for the timeout. The default value must be changed to a value not equal to zero, so that the watchdog can be started. The time is set in multiples of 100 ms, e.g., 0x0009 means a timeout time of 0.9 seconds. This value cannot be changed when the watchdog is running. There is no code by which the current data value can be written again while the watchdog is active.

Table 64: Register address 0x1001

Register address 0x1001 (4097 _{dec})	
Value	Watchdog function coding screen, function code 1...16
Access	Read/write
Default	0x0000
Description	Use this screen to set the function codes to trigger the watchdog function. The function code can be selected with "1". $(= 2^{(\text{function code}-1)} + 2^{(\text{function code}-1)} + \dots)$ Example: Function code 5 $\rightarrow 2^{(5-1)} = 2^4 \rightarrow$ bit 4 is set to 1 A value not equal to zero starts the watchdog function. If only codes from unsupported functions are entered in the screen, the watchdog does not start. An existing error is reset and the process image can be described again. Again here, no changes can be made when the watchdog is running. When the watchdog is active, there is no code by which the current data value can be written again.

Table 65: Register address 0x1002

Register address 0x1002 (4098 _{dec})	
Value	Watchdog function coding screen, function code 17...32
Access	Read/write
Default	0x0000
Description	The same function as before, but with the function codes 17 to 32. These codes are not supported. Therefore, this register should be left at the default value. There is no exception code by which the current data value can be written again while the watchdog is active.

Table 66: Register address 0x1003

Register address 0x1003 (4099 _{dec})	
Value	Watchgod trigger
Access	Read/write
Default	0x0000
Description	This register is used for an alternative trigger method. The watchdog is triggered by writing different values to this register. Successive values must differ in size. The watchdog starts when values not equal to zero are written. A watchdog error is reset and it is again possible to write process data.

Table 67: Register address 0x1004

Register address 0x1004 (4100 _{dec})	
Value	Minimum current trigger time
Access	Read/write
Default	0xFFFF
Description	This register saves the current smallest watchdog trigger time. When the watchdog is triggered, the saved value is compared to the current value. If the current value is smaller than the saved value, it is replaced by the current value. The unit is 100 ms/digit. The saved value is modified by writing new values. This has no effect on the watchdog. The value 0x0000 is not permitted.

Table 68: Register address 0x1005

Register address 0x1005 (4101 _{dec})	
Value	Stop watchdog
Access	Read/write
Default	0x0000
Description	If the value 0xAAAA followed by the value 0x5555 is written to this register, the watchdog stops. The watchdog error response is blocked. A watchdog error is reset and it is again possible to write to the process data.

Table 69: Register address 0x1006

Register address 0x1006 (4102 _{dec})	
Value	While watchdog runs
Access	Read
Default	0x0000
Description	Current watchdog status at 0x0000: Watchdog inactive at 0x0001: Watchdog active at 0x0002: Watchdog timed out

Table 70: Register address 0x1007

Register address 0x1007 (4103 _{dec})	
Value	Restart watchdog
Access	Read/write
Default	0x0001
Description	Writing 0x1 to the register starts the watchdog again. If the watchdog was stopped before the overflow, it is not started again.

Table 71: Register address 0x1008

Register address 0x1008 (4104 _{dec})	
Value	Just pause watchdog
Access	Read/write
Default	0x0000
Description	By writing the values 0x0AA55 or 0X55AA, the watchdog is paused if active. The watchdog error response is temporarily disabled. An existing watchdog error is reset and it is again possible to write to the watchdog register.

Table 72: Register address 0x1009

Register address 0x1009 (4105 _{dec})	
Value	Close MODBUS socket after watchdog timeout
Access	Read/write
Description	0: MODBUS socket is not closed 1: MODBUS socket is closed

Table 73: Register address 0x100A

Register address 0x100A (4106 _{dec})	
Value	Alternative watchdog
Access	Read/write
Default	0x0000
Description	Write a time value to register 0x1000 Register 0x100A = 0x0001: Watchdog is switched to active The watchdog is started with the first MODBUS telegram. Each MODBUS/TCP command triggers the watchdog. All outputs are set to zero after the watchdog time has elapsed. The outputs can be set again by writing again. The register 0x100A is permanent and also register 0x1000. When the watchdog is on, the time value in register 0x1000 can no longer be modified.

Table 74: Register address 0x100B

Register address 0x100B (4107 _{dec})	
Value	Save Watchdog Parameters
Access	Write
Default	0x0000
Description	By writing "1" to register Register 0x100B, registers 0x1000, 0x1001, 0x1002 are set to permanent.

13.3.4.2 Diagnostics Registers

The following registers can be read to determine an error in the fieldbus node.

Table 75: Register address 0x1020

Register address 0x1020 (4128 _{dec})	
Value	LED Error Group
Access	Read
Description	Indication of the group number

Table 76: Register address 0x1021

Register address 0x1021 (4129 _{dec})	
Value	LED Error Code/Argument
Access	Read
Description	Indication of the error code and error argument

13.3.4.3 Configuration Registers

The following registers can be read to determine the configuration of the connected I/O modules:

Table 77: Register address 0x1022

Register address 0x1022 (4130 _{dec})	
Value	CnfLen.AnalogOut
Access	Read
Description	Number of I/O bits for the process data words of the outputs

Table 78: Register address 0x1023

Register address 0x1023 (4131 _{dec})	
Value	CnfLen.AnalogInp
Access	Read
Description	Number of I/O bits for the process data words of the inputs

Table 79: Register address 0x1024

Register address 0x1024 (4132 _{dec})	
Value	CnfLen.DigitalOut
Access	Read
Description	Number of I/O bits for the process data bits of the outputs

Table 80: Register address 0x1025

Register address 0x1025 (4133 _{dec})	
Value	CnfLen.DigitalInp
Access	Read
Description	Number of I/O bits for the process data bits of the inputs

Table 81: Register address 0x1028

Register address 0x1028 (4136 _{dec})	
Value	Boot options
Access	Read/write
Description	Boot configuration: 1: BootP 2: DHCP 4: EEPROM

Table 82: Register address 0x1029

Register address 0x1029 (4137 _{dec}) with up to 9 words	
Value	MODBUS/TCP statistics
Access	Read/write
Description	1 word SlaveDeviceFailure → Terminal bus error, fieldbus error with watchdog switched on 1 word BadProtocol → Error in MODBUS/TCP header 1 word BadLength → Incorrect telegram length 1 word BadFunction → Invalid function code 1 word Bad Address → Invalid register address 1 word BadData → Invalid value 1 word TooManyRegisters → Number of registers to be processed too high, read/write 125/100 1 word TooManyBits → Number of coils to be processed too high, read/write 2000/800 1 word ModTcpMessageCounter → Number of MODBUS/TCP telegrams received The register is reset by writing 0xAA55 or 0x55AA.

Table 83: Register address 0x102A

Register address 0x102A (4138 _{dec}) with up to 1 word	
Value	MODBUS/TCP connections
Access	Read
Description	Number of TCP connections

Table 84: Register address 0x1030

Register address 0x1030 (4144 _{dec}) with up to 1 word	
Value	Configuration MODBUS/TCP timeout
Access	Read/write
Default	0x0000
Description	This register saves the value for TCP connection monitoring. The time base is 100 ms, the minimum value is 10 ms. Open TCP connections are automatically closed if the time entered was exceeded per connection. A query on the connection triggers the watchdog.

Table 85: Register address 0x1031

Register address 0x1031 (4145 _{dec}) with up to 3 words	
Value	Read the MAC ID of the fieldbus coupler
Access	Read
Description	Output of the MAC ID, length 3 words

Table 86: Register address 0x2030

Register address 0x2030 (8240 _{dec}) with up to 65 words	
Value	Description of the connected I/O module
Access	Read I/O modules 0...64
Description	Output: e.g., "3801" for the input module 767-3801

Table 87: Register address 0x2040

Register address 0x2040 (8256 _{dec})	
Value	Perform software reset
Access	Write (write sequence 0xAA55 or 0x55AA)
Description	The fieldbus coupler performs a restart by writing the values 0xAA55 or 0x55AA.

Table 88: Register address 0x2041

Register address 0x2041 (8257 _{dec})	
Value	Flash format
Access	Write (write sequence 0xAA55 or 0x55AA)
Description	The Flash file system is reformatted.

Table 89: Register address 0x2042

Register address 0x2042 (8258 _{dec})	
Value	Extract files
Access	Write (write sequence 0xAA55 or 0x55AA)
Description	The default files (HTML pages) of the fieldbus coupler are extracted and written to the Flash file system.

Table 90: Register address 0x2043

Register address 0x2043 (8259 _{dec})	
Value	0x55AA
Access	Write
Description	Default settings

13.3.4.4 Firmware Information Register

The following registers are used to read information about the fieldbus coupler firmware.

Table 91: Register address 0x2010

Register 0x2010 (8208_{dec}) with up to 1 word	
Value	Revision
Access	Read
Description	Firmware index, e.g., 0005 for version 5

Table 92: Register address 0x2011

Register address 0x2011 (8209_{dec}) with up to 1 word	
Value	Series code
Access	Read
Description	WAGO series number, e.g., 0767 for WAGO SPEEDWAY 767

Table 93: Register address 0x2012

Register address 0x2012 (8210_{dec}) with up to 1 word	
Value	Item number
Access	Read
Description	WAGO item number, e.g., 2301 for the fieldbus coupler 767-2301, etc.

Table 94: Register address 0x2013

Register address 0x2013 (8211_{dec}) with up to 1 word	
Value	Major sub item code
Access	Read
Description	Firmware version major revision

Table 95: Register address 0x2014

Register address 0x2014 (8212 _{dec}) with up to 1 word	
Value	Minor sub item code
Access	Read
Description	Firmware version minor revision

Table 96: Register address 0x2020

Register address 0x2020 (8224 _{dec}) with up to 16 words	
Value	Description
Access	Read
Description	Information about the fieldbus coupler, 16 words

Table 97: Register address 0x2021

Register address 0x2021 (8225 _{dec}) with up to 8 words	
Value	Description
Access	Read
Description	Time of the firmware version, 8 words

Table 98: Register address 0x2022

Register address 0x2022 (8226 _{dec}) with up to 8 words	
Value	Description
Access	Read
Description	Date of the firmware version, 8 words

Table 99: Register address 0x2023

Register address 0x2023 (8227 _{dec}) with up to 32 words	
Value	Description
Access	Read
Description	Information about programming the firmware, 32 words

13.3.4.5 Constant Register

The following registers contain constants that can be used to test communication with the master:

Table 100: Register address 0x2000

Register address 0x2000 (8192 _{dec})	
Value	Zero
Access	Read
Description	Constant with zero

Table 101: Register address 0x2001

Register address 0x2001 (8193 _{dec})	
Value	Ones
Access	Read
Description	Constant with ones. • "-1" if constant is declared as "signed int" • "MAXVALUE" if constant is declared as "unsigned int"

Table 102: Register address 0x2002

Register address 0x2002 (8194 _{dec})	
Value	1,2,3,4
Access	Read
Description	Constant value for testing whether high and low bytes are exchanged (Intel/Motorola format). Should appear in the master as 0x1234. If 0x3412 appears, high and low bytes must be exchanged.

Table 103: Register address 0x2003

Register address 0x2003 (8195 _{dec})	
Value	Screen 1
Access	Read
Description	Constant that displays whether all bits are present. Used together with register 0x2004.

Table 104: Register address 0x2004

Register address 0x2004 (8196 _{dec})	
Value	Screen 1
Access	Read
Description	Constant that displays whether all bits are present. Used together with register 0x2003.

Table 105: Register address 0x2005

Register address 0x2005 (8197 _{dec})	
Value	Largest positive number
Access	Read
Description	Constant to check the arithmetic

Table 106: Register address 0x2006

Register address 0x2006 (8198 _{dec})	
Value	Largest negative number
Access	Read
Description	Constant to check the arithmetic

Table 107: Register address 0x2007

Register address 0x2007 (8199 _{dec})	
Value	Largest half positive number
Access	Read
Description	Constant to check the arithmetic

Table 108: Register address 0x2008

Register address 0x2008 (8200 _{dec})	
Value	Largest half negative number
Access	Read
Description	Constant to check the arithmetic

14 CoDeSys 3 Runtime Environment

This section contains information on configuring the gateway and using CoDeSys (WAGO-OEM Setup, item no. 759-915) to create a project.

14.1 Installing the CoDeSys Programming System

CoDeSys 3 is based on Microsoft Plug-In technology, whereby different versions (profiles) of CoDeSys can be installed in one directory. In the first installation, the Microsoft Framework 2.0 and all plug-ins associated with this profile are installed. For each subsequent installation, merely a new profile is created and any new plug-ins registered.

The following WAGO-specific expansions are also included in the installation:

- Project templates
- Device description
- Libraries
- USB drivers

The WAGO project templates already contain the most important elements of an application program and save time when creating a new project. The device descriptions contain all device-specific information for the WAGO 767 product series.

To install the CoDeSys 3 programming software on the fieldbus coupler, please proceed as follows.

1. Insert the CoDeSys 3 CD into your CD-ROM drive.
2. To install the programming system, follow the instructions that appear on your screen. A successful installation is indicated by a CoDeSys icon on your desktop.

14.2 Editing the Gateway Configuration for CoDeSys 3

Requirement: The fieldbus coupler must be switched on and your PC must be connected with the USB interface. For more information, see Section 8.7.

To edit the gateway for serial communication, proceed as follows:

1. Double-click on the "Remove Hardware" icon in the taskbar to display the existing COM ports.



Figure 39: "Remove Hardware" icon in the taskbar

Assigned COM ports are not displayed unless the "Display Device Components" selection box has been selected.

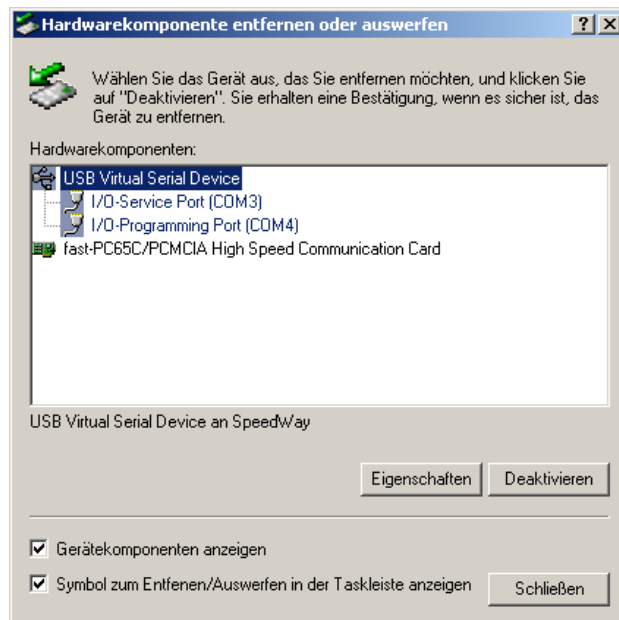


Figure 40: "Remove or Eject Hardware Components" dialog box

2. Note the COM port of the "I/O-Programming" gateway. This is COM port 4 in this example.
3. Close the dialog box by clicking [Close].



Note

This COM port number remains constant for the fieldbus coupler currently connected and for the USB interface currently being used. When you replace the fieldbus coupler or the USB slot on your PC, the COM port also changes.

After installing CoDeSys, you must edit the ComPort= entry in the "Gateway.cfg" configuration file to conform to the local requirements.

To do this, open the "Edit Gateway.cfg" link that was created on the desktop during CoDeSys installation.



Figure 41: Edit Gateway.cfg

If this link is not available on your desktop, the configuration file can typically be found under **Programs > 3S > CoDeSys > GatewayPLC**. Open the Gateway.cfg configuration file with an editor.

1. Prefer to section [CmpBlkDrvSimpleCom] for ComPort=.

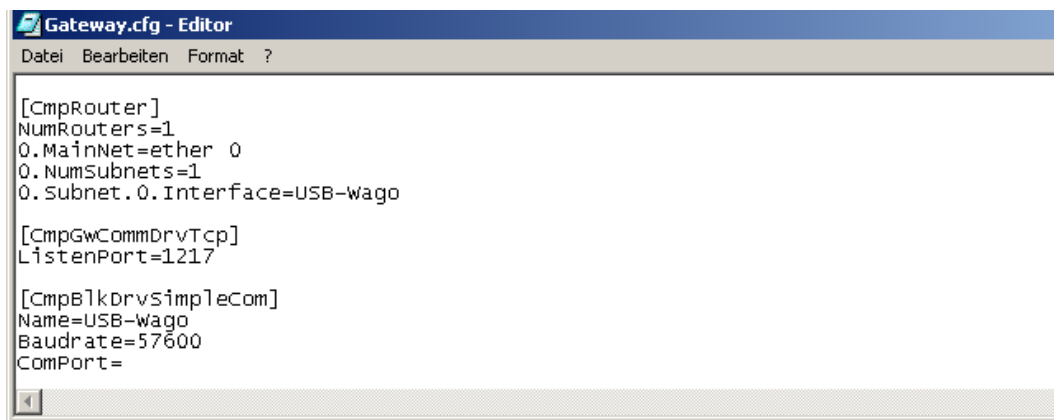


Figure 42: "Gateway.cfg" configuration file

2. Enter the COM port number of the connected fieldbus coupler after ComPort=. In this example, the number is 4.

```

[CmpBlkDrvSimpleCom]
Name=USB WAGO
Baudrate=57600
ComPort=4
  
```

3. Overwrite the modified configuration file as follows: Select **File > Save As...** The **Save File As** dialog box opens. In the **File Type** field, select "All files" to maintain the ".cfg" file ending. Click [**Save**] to overwrite.

14.3 Programming with CoDeSys 3

This Section uses an example of the WAGO project template to explain the relevant steps in creating a CoDeSys project.

This Section is meant to be a quick start guide and does not cover the entire scope of CoDeSys functions. A detailed description of all features can be found on the CoDeSys CD-ROM (759-915), obtainable from WAGO.

14.3.1 Start the CoDeSys Programming System

To start CoDeSys, double-click on the CoDeSys icon on your desktop or navigate to the program via your operating system's Start menu as follows:

Start > Programs > 3S CoDeSys > CoDeSys > CoDeSys V3.

14.3.2 Creating a CoDeSys Project

In the WAGO-specific CoDeSys installation, a project template already exists in which the settings are selected in such a way that a PLC program can be developed on the fieldbus coupler.

1. In the menu bar, click on **File** and select **New Project...**. The "New Project" dialog box opens.
2. To open the WAGO project template, click on the "WAGO" folder in the "Categories" field.
3. Click on the "767-2301 Project" icon in the "Templates" field.
4. In the "Name" input field, enter a project name.
5. In the "Location" input field, indicate the location where you wish to save the project by clicking on the [...] button.
6. Click **[OK]** to confirm your entry.

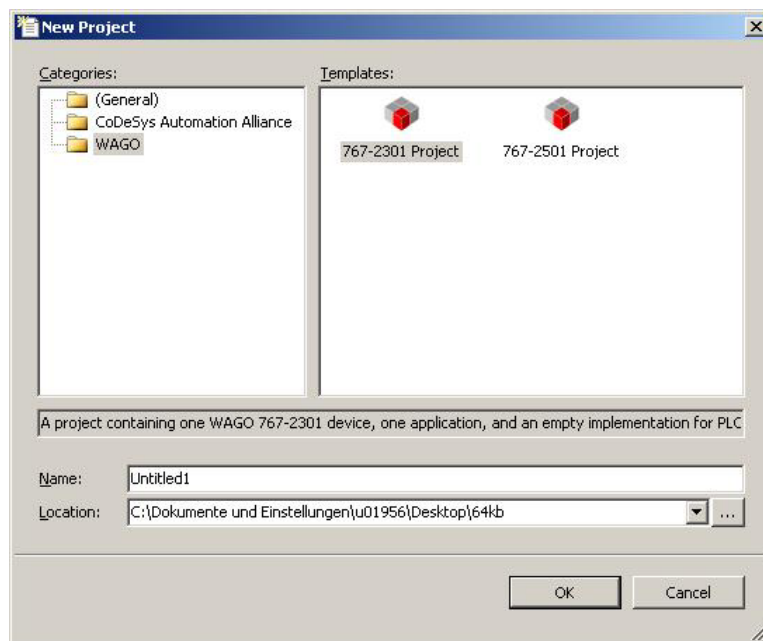


Figure 43: "New Project" dialog box

You can achieve the same results if you begin with a blank project and then manually add the necessary objects.

14.3.3 Adding Library Management

In the library management, all libraries are listed whose function blocks can be used in your project.

You can also use other library management objects to more precisely define the application area of the libraries. For the sake of simplicity, we will confine ourselves for this example to a general library management. If needed, use the online assistance or the manual for CoDeSys 3.

1. In the "POUs" window, right-click on the project name that you have assigned. In this example, the name is "FirstProject".
2. In the context menu, select **Add Object...**. The "Add Object" dialog box opens.

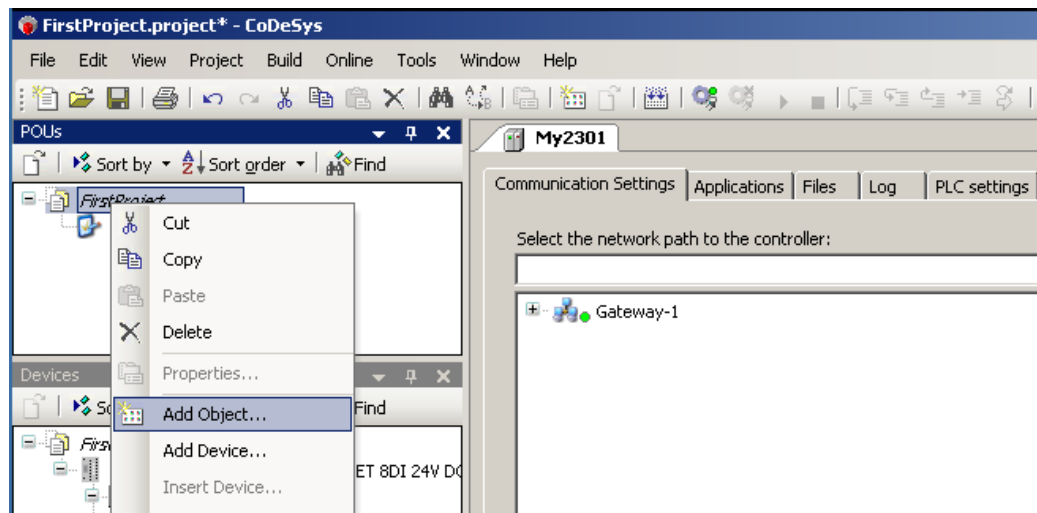


Figure 44: "Add Object" dialog box

3. In the "Add Object" dialog box, click on the "Library Manager" option.
4. Click **[Open]** to add the library manager.

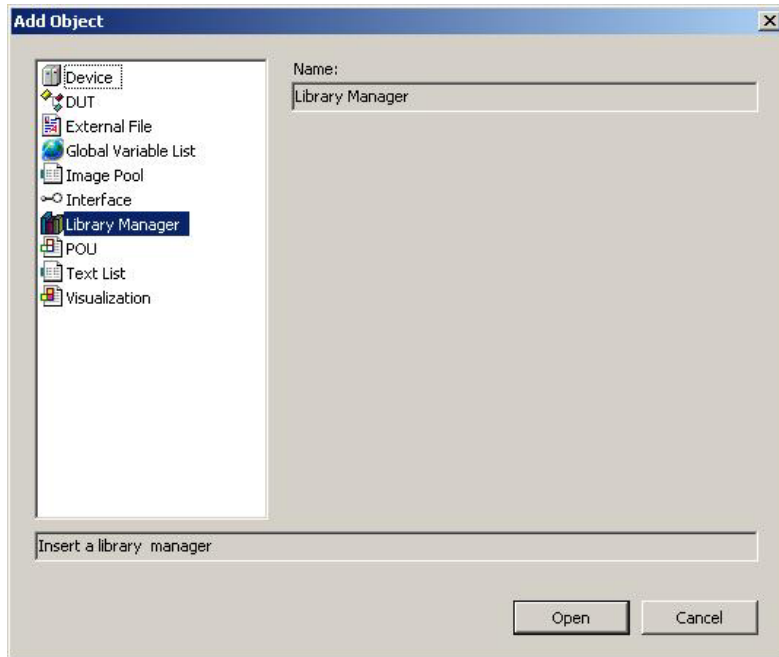


Figure 45: "Add Object" dialog box with the "Library Manager" selection

5. The library management opens.

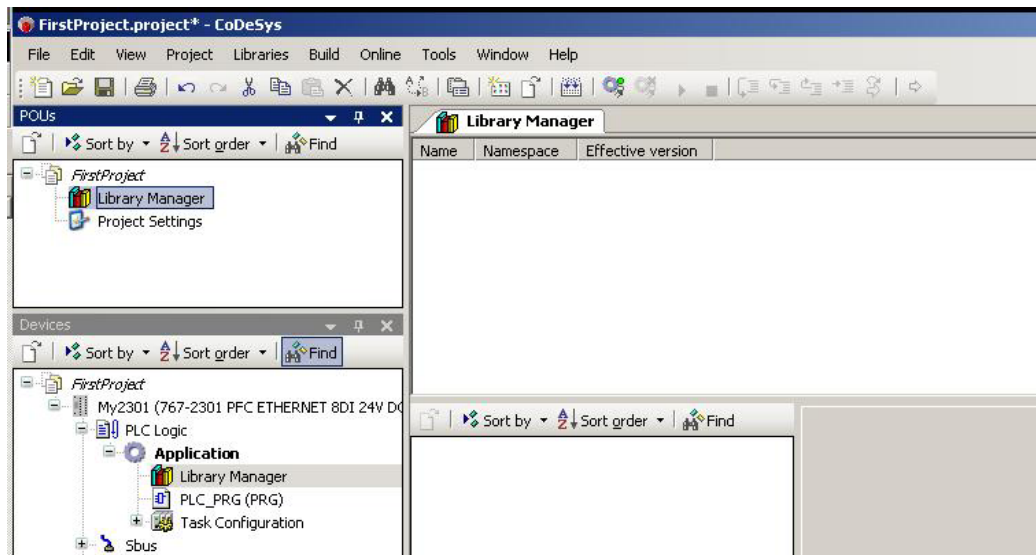


Figure 46: "Library Manager" view

14.3.4 Adding Libraries

In order to use the function blocks of a library in the PLC program, you must add this library in the library management.

Note



In order to use a function block whose library has not yet been referenced, this step is carried out during program processing.

To add a reference to the "Standard" library in the library management, proceed as follows:

1. Click on **Libraries** in the menu bar and select **Add Library**. The "Add Library" dialog box opens.
2. In the "Add Library" dialog box, click on "Common" and select the "Standard" library.
3. Click **[OK]** to submit your selection.

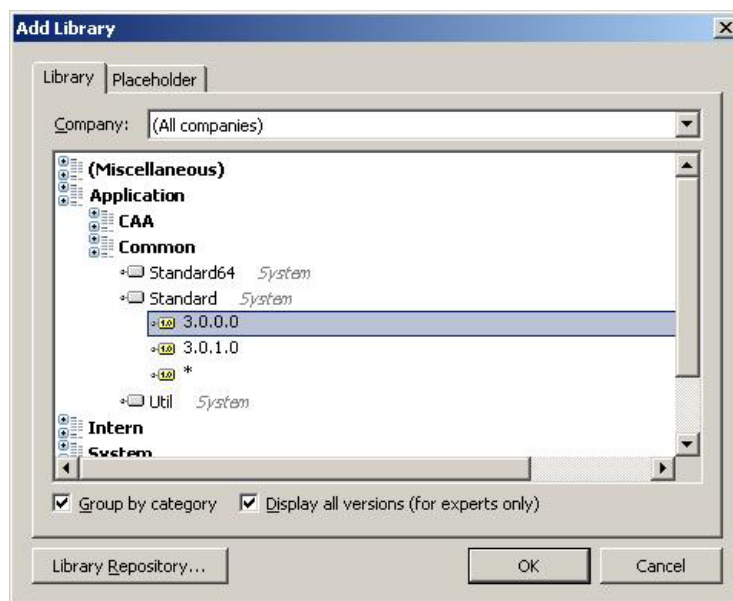


Figure 47: "Add Library" dialog box with the "Standard" selection

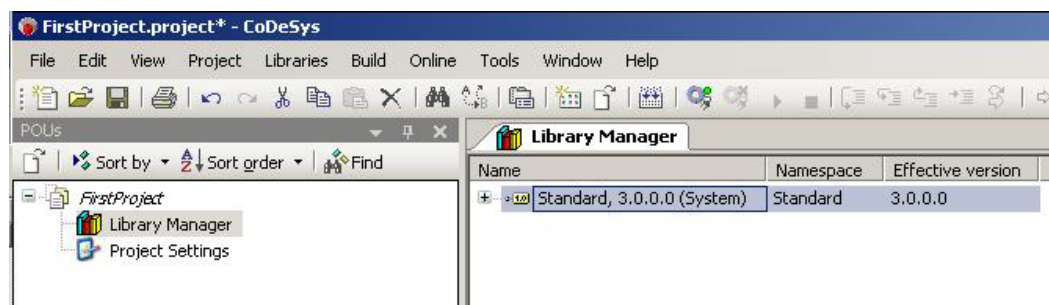


Figure 48: "Library Manager" view with the "Standard 3.0.0.0" library.

14.3.5 Creating a Control Program

A control program (POU) needs to be created for the project. POU stands for "Program Organizational Unit". This control program is created in an IEC 6131-3 programming language (AWL, FUP, etc.)

In the WAGO project template, a PLC program already exists with the name PLC_PRG (PRG). This program will be used for this description.

1. Right-click on the project name that you have assigned. In this example, the name is "FirstProject".
2. In the context menu, select **Add Object...**. The "Add Object" dialog box opens.

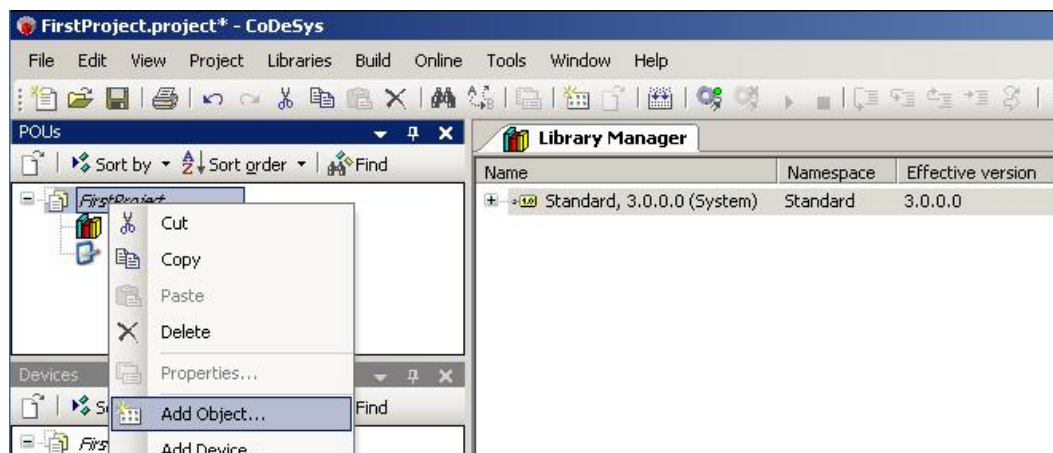


Figure 49: Creating a control program

3. In the "Add Object" dialog box, click on the "POU" object.
4. In the "Name" field, enter a name for your control program; e.g., FirstProgram.
5. In the "Type" field, select the **Program** option.
6. From the "Implementation Language" menu, select a programming language in which the PLC program should be created.
7. Click **[Open]** to confirm your selection.

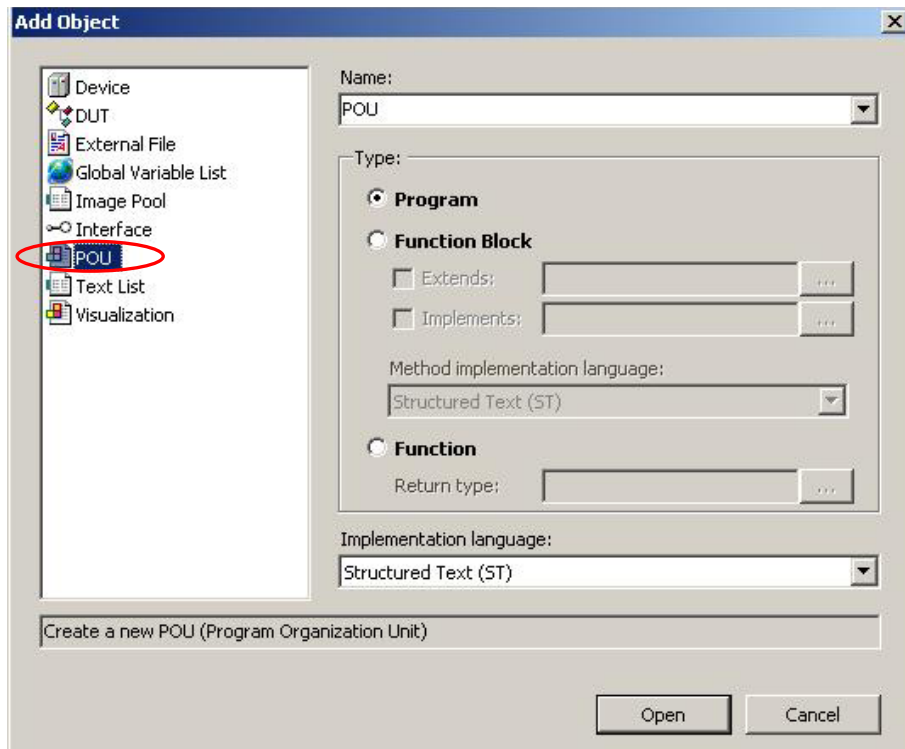


Figure 50: "Add Object" dialog box with the "POU" selection

8. To open the editor of your selected programming language, double-click on your program names. For further questions on Semantik, use the online assistance or the manual for CoDeSys 3.x.

14.3.6 Creating a Task Configuration

The task configuration is used to configure all tasks of the controller.

A task configuration (named "Task Configuration") already exists in the WAGO project template. This Section is therefore only necessary if you wish to add another controller in your project. This section is only necessary if you wish to create a new project not based on the WAGO project template.

A task management object is created as follows:

1. Right-click on "Application" in the "Devices" window. In the context menu, select "Add Object". The "Add Object" dialog box appears.

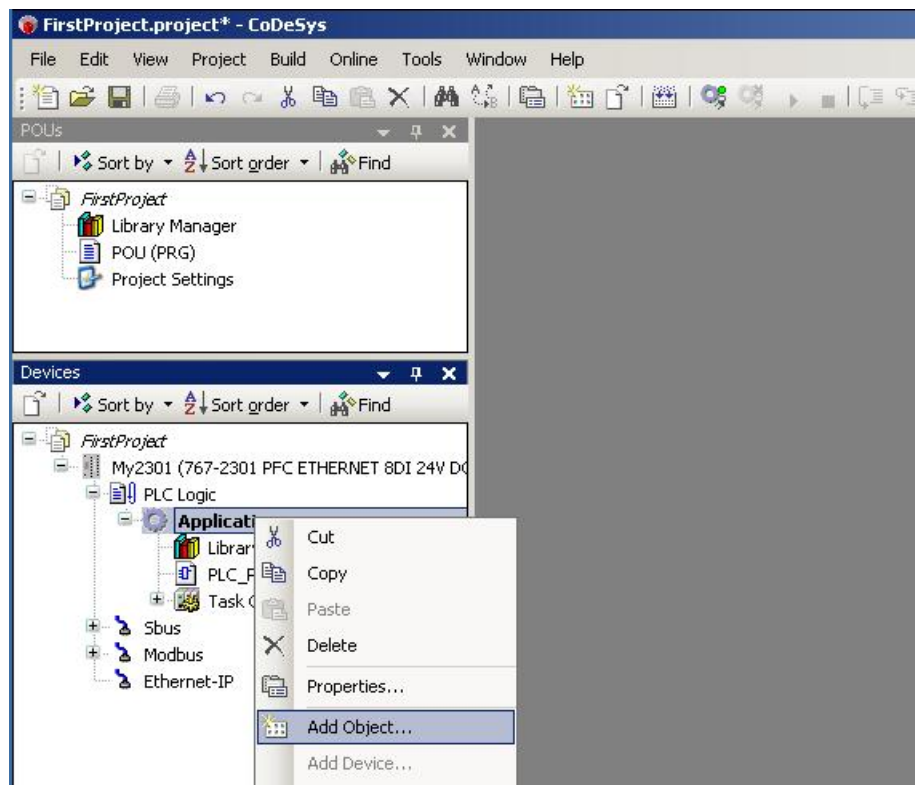


Figure 51: Creating a task management object

2. In the "Add Object" dialog box, select the "Task Configuration" object type.
3. Click **[Open]** to confirm your selection.

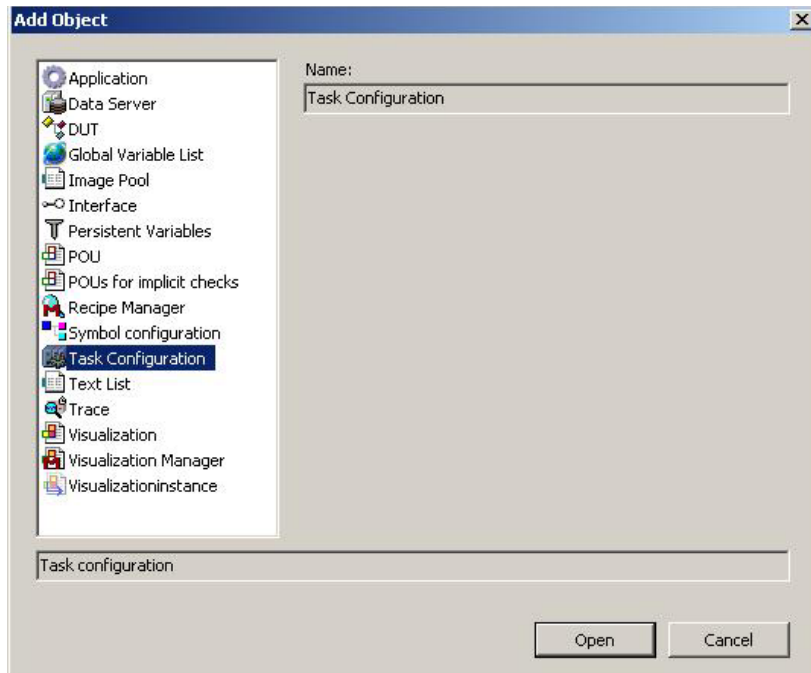


Figure 52: "Add Object" dialog box with the "Task Configuration" selection

14.3.6.1 Creating a Task

Create the task that will later access programs which are to be executed by the fieldbus coupler. A task can trigger program access in either a cyclical or an event-oriented manner.

Divide your automation task into several tasks with differing cycle times in order to complete automation tasks at different times.

1. To create a task, right-click on "Task Configuration" in the "Devices" window.
2. In the context menu, select **Add Object....** The "Add Object" dialog box appears.

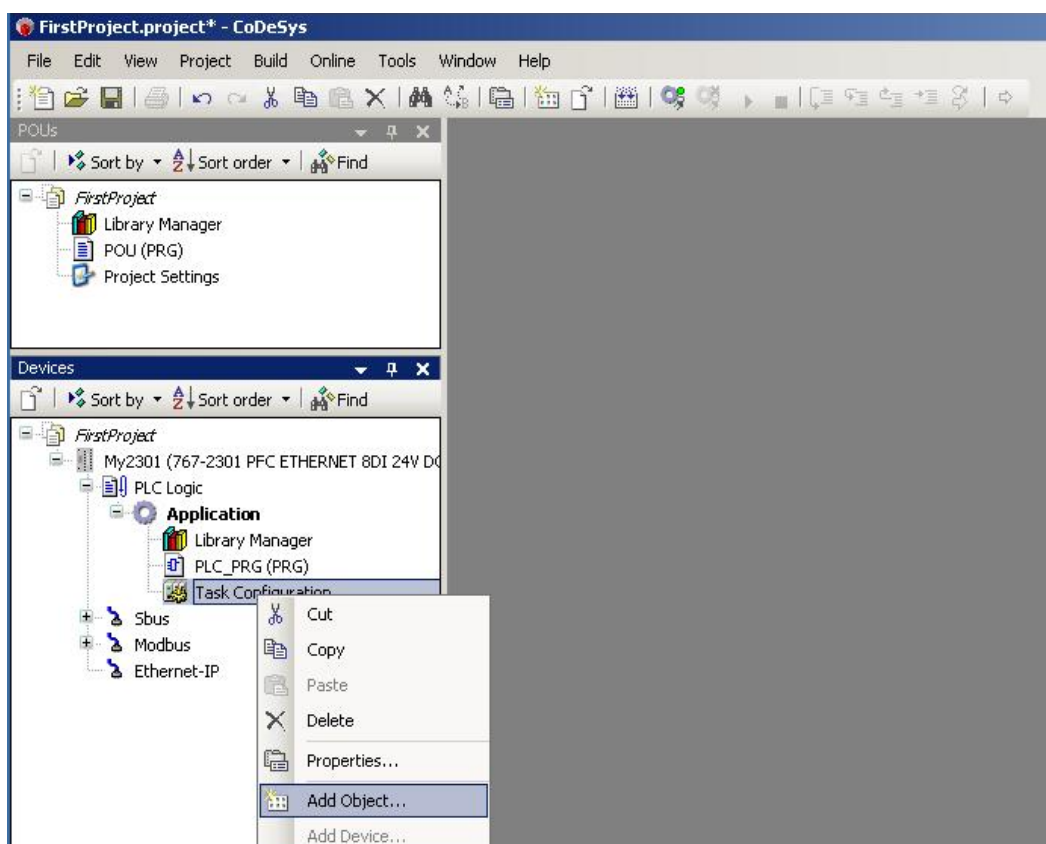


Figure 53: Creating a task

Note



The fieldbus coupler controller provides integrated monitoring of the CPU load. In case of an overload, the current application is closed. This exception is recorded in the "Log" tab under "My2301." The entry reads: `"*Exception* WatchdogApplication<TaskName>."`

3. Rename the task by assigning it a different name in the "Name" input field. In this example, the name is "myTask".
4. Click [**Open**] to submit your entry.

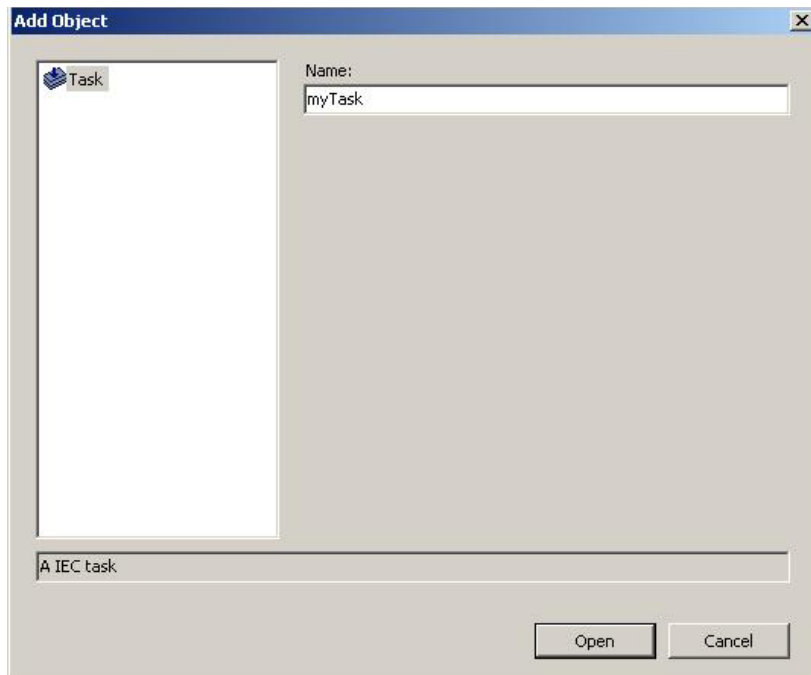


Figure 54: "Add Object" dialog box

14.3.6.2 Assignment between POU and Task

In the WAGO project template, the "PLC_PRG" POU is already added to a call task. A POU can be executed in several controllers. Ensure that the environmental parameters (e.g., comprehensive variables) of the various controllers are defined identically. The following steps are only relevant if you want to add another POU as call.

1. To open the task configuration input window, right-click on "myTask" in the "Device" pane.
2. Add a program to the task by clicking on the [Add POU] button. The "Input Assistance" dialog box appears.

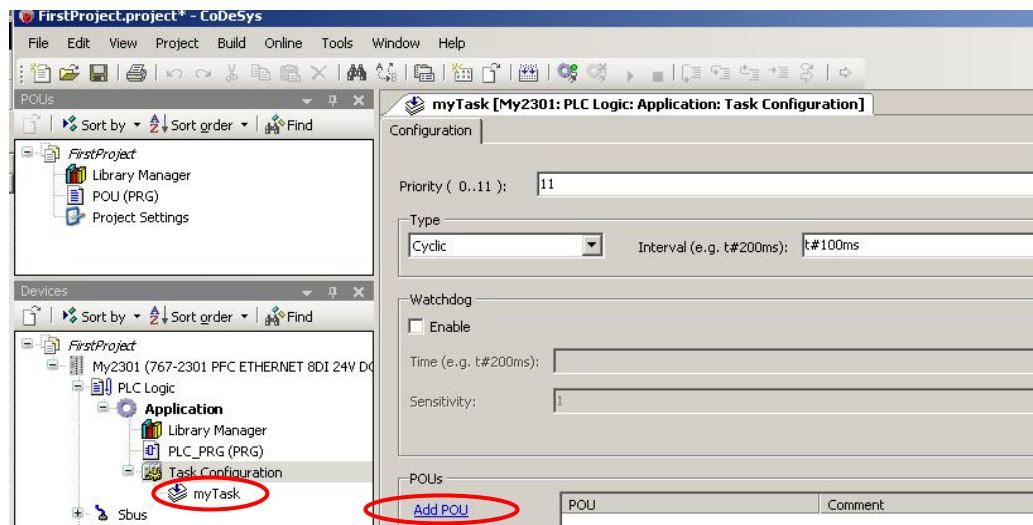


Figure 55: Assignment between POU and task

3. In the "Input Assistance" dialog box, select the "Program (Project)" option. In the "Elements" field, select the POU that should be added.

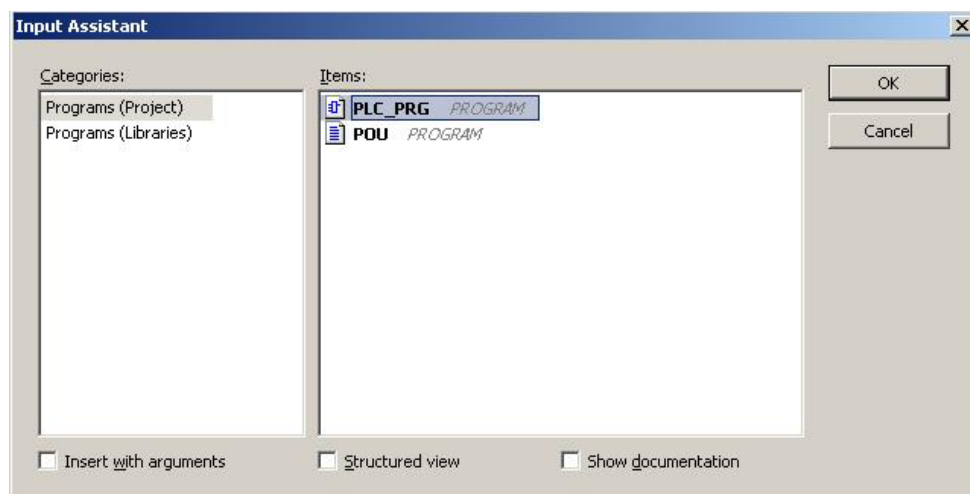


Figure 56: "Input Assistance" dialog box

4. Click [OK] to submit your entry.

14.3.7 Selecting the Communication Setting

Select here the controller (fieldbus coupler) on which your current project is to be executed. Connect the fieldbus coupler to your PC. The USB cable can be used for this (see Section 8.7).

Communication between CoDeSys and the fieldbus coupler is established by the CoDeSys gateway. The gateway is installed together with the installation of CoDeSys. The gateway icon appears in the system tray.



Figure 57: System tray on your desktop

1. Double-click on your device description in the "Devices" window. The controller description window appears. The gateway can be seen in the "Communication Settings" tab.
2. Browse the network for the fieldbus coupler by selecting "Gateway" and clicking on **[Scan network]**.

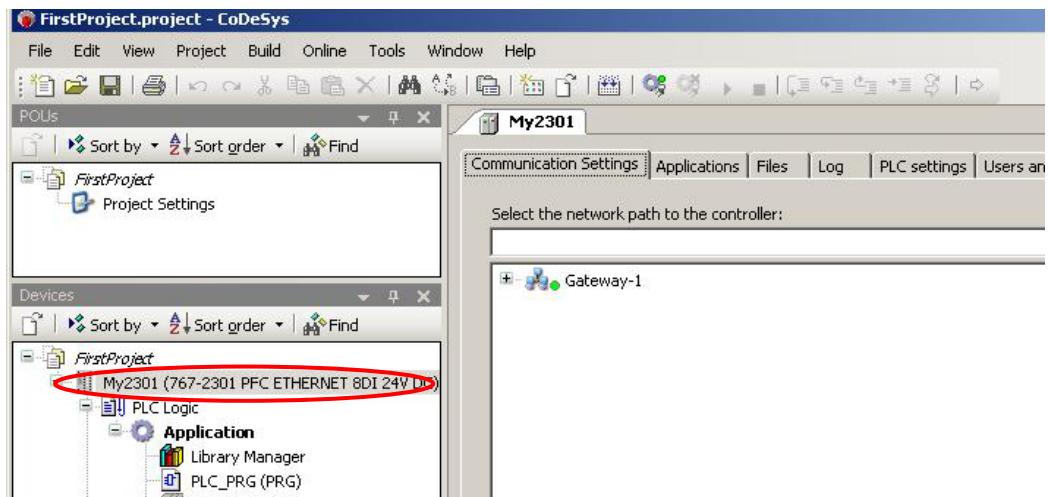


Figure 58: "Communication Settings" tab

3. Select the fieldbus coupler and click on **[Set active path]** to assign this to the configured fieldbus coupler.

14.3.8 Loading a PLC Program

CoDeSys is capable of managing several fieldbus couplers (controllers) in a project. Only one controller at a time can be loaded and started. Thus, you must define which application should be loaded. For the sake of simplicity, this example has only one application and one controller. Thus, the application in this example is always the "active application".

To load a PLC program, proceed as follows:

1. In the "Device" pane, right-click on the "Application" that you want to load, and select **Set Active Application** from the context menu.

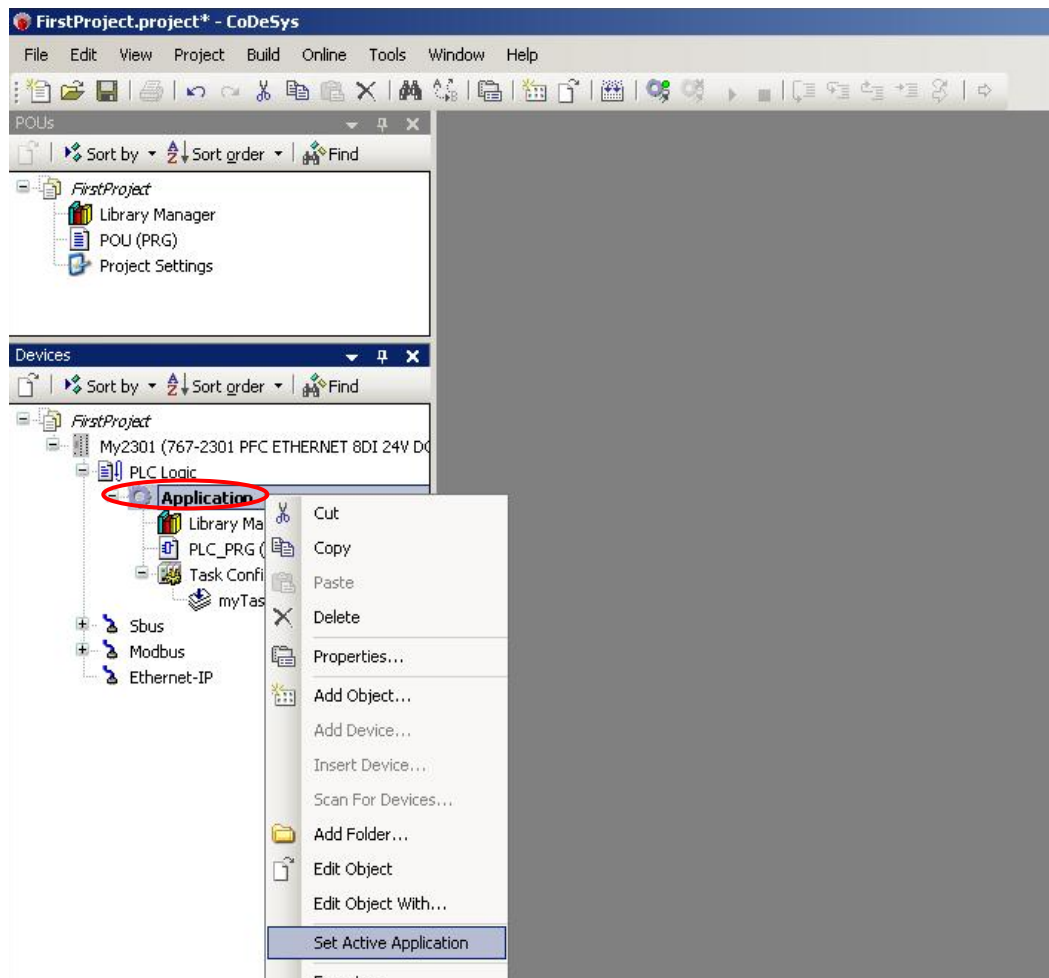


Figure 59: Loading a PLC program

The "Decode" and "Load" functions work on the "active application" of a controller. All programs that are assigned to the application by a task are decoded and loaded. Programs that are not assigned to a task are not decoded or loaded.

2. To load, click on **Online > Login** from the menu bar. This automatically generates the binary code for the controller. The following confirmation prompt appears. Click **[Yes]** to confirm.



Figure 60: Confirmation prompt

Note



To load the program, it must be decoded without errors. If errors exist in the program, click on the error message and correct the error. If needed, use the online assistance or the manual for CoDeSys 3.

3. To start the application that you have just loaded into the controller, select **Online > Start** from the menu bar.

In the online mode of the editor, you can view the online data of the PLC program and change the data, if needed. You can also disrupt the operation of the PLC program via breakpoints and proceed through the PLC program in individual steps. Additional information in this regard can be found in the online assistance or in the CoDeSys 3 manual.

4. Select **Online > Logout** from the menu bar.
5. To retain your intact application in the fieldbus coupler, select **Online > Create boot application for "Application"...**

14.3.9 Adding I/O Modules

To access the process data of the I/O modules, you must first insert the I/O modules being used into the controller configuration. This does not require that you configure the entire physical structure of the 767 node. Simply establish the desired I/O modules in the CoDeSys project.

To attach the I/O modules, please proceed as follows:

1. In the "Devices" window, right-click on "Sbus" and select **Add Device...** from the context menu. The "Add Device" dialog box opens

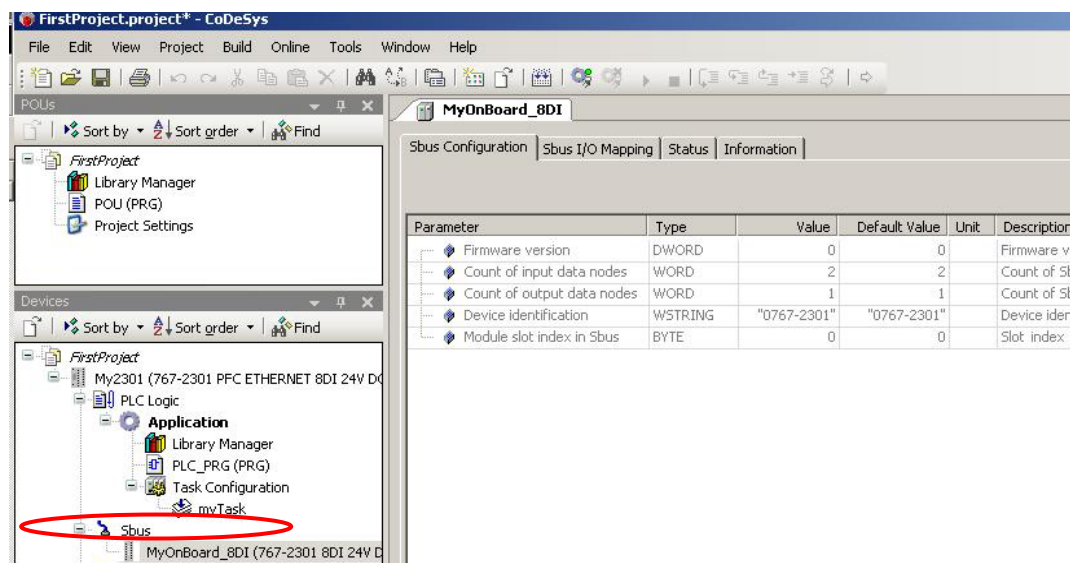


Figure 61: Adding I/O modules

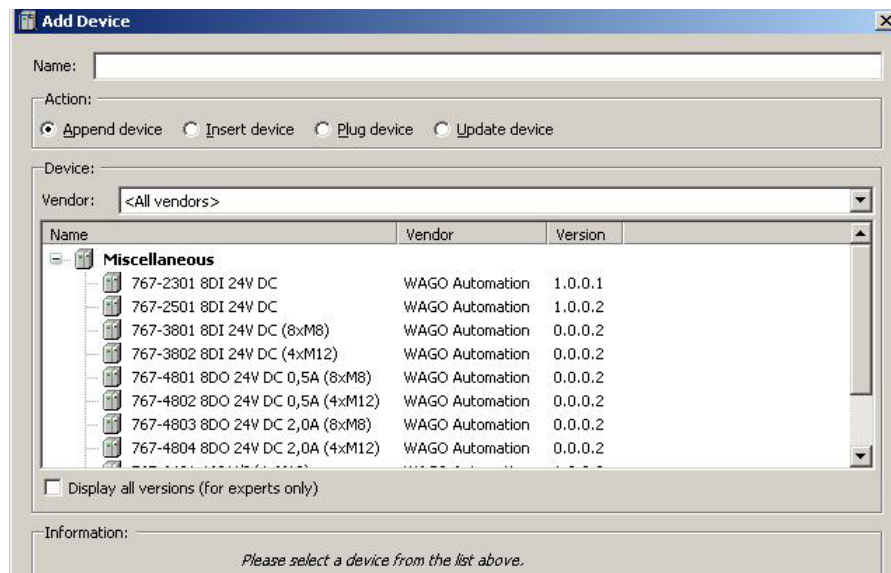


Figure 62: "Attach Device" dialog box

2. Choose the I/O module that you wish to configure by selecting the respective I/O module and clicking on **[Add Device]**. Click **[Close]** to close the dialog box.

3. To clearly define the inserted I/O module, you must assign a clear number to the "Module Slot Index in Sbus" parameter in the "SBUS Configuration" tab. This number indicates the position of the I/O module in the 767 node. For example, the number 4 is entered to access the fourth I/O module in the 767 node.

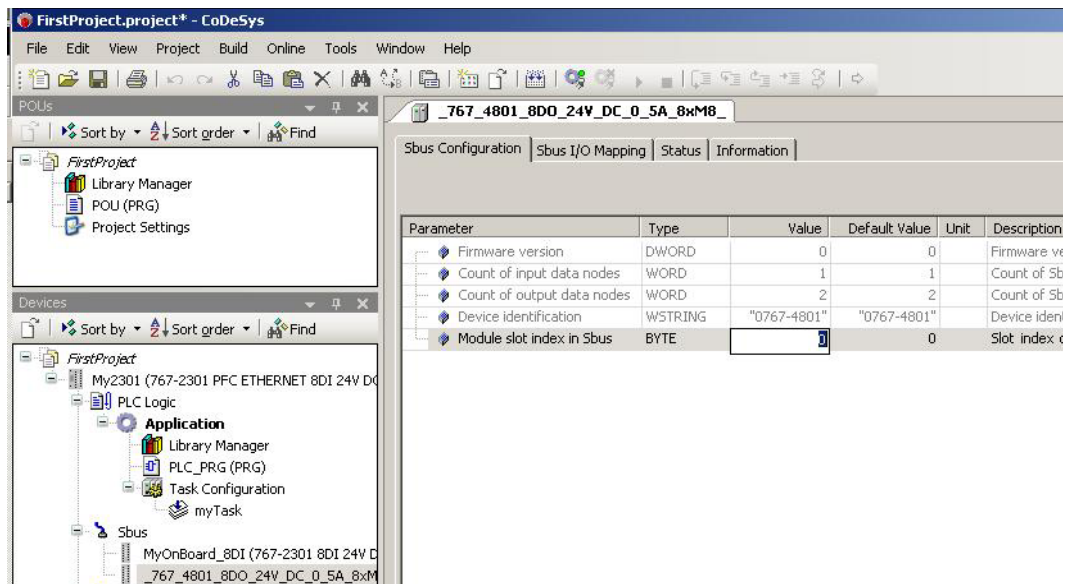


Figure 63: Selection of I/O modules in "Attach Device" dialog box

14.3.9.1 Assignment Between the Process Data of the I/O Modules and the Variables of the Control Program

In order to access the process data of the I/O modules, you must connect them with a variable. You have two options:

- Link pre-defined variables with the I/O modules
- Define new variables

Linking pre-defined variables with the fieldbus variables

1. To link variables, click on the "SBUS I/O Image" tab and open the "Output Channels" tree.
2. Double-click in the "Mapping" pane to choose between the options "Map to Existing Variable" and "Create New Variable." Select "Map to Existing Variable."
3. In the "Variable" pane, enter the name and path of a variable that you have created in CoDeSys. The following variable is contained in the WAGO project template: Application.PLC_PRG.iCount.

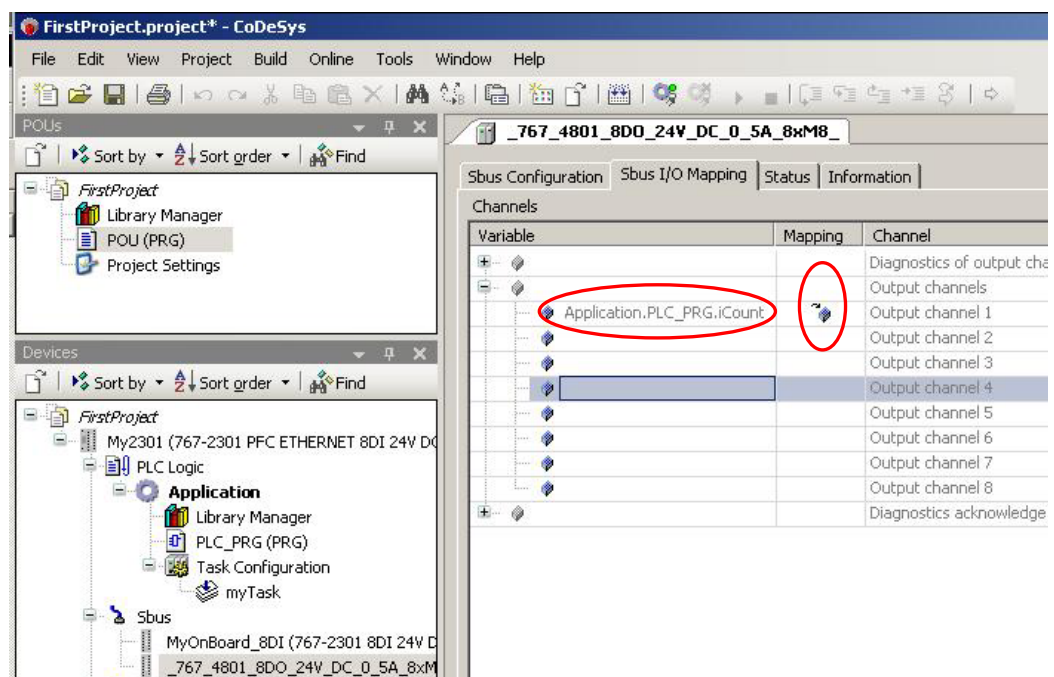


Figure 64: Link variables used in the program using inputs and outputs

Defining New Variables

1. To define variables, click on the "SBUS I/O Image" tab and open the "Output Channels" tree.
2. Double-click on the icon in the "Mapping" pane to choose between the options "Map to Existing Variable" and "Create New Variable." Select "Create New Variable."

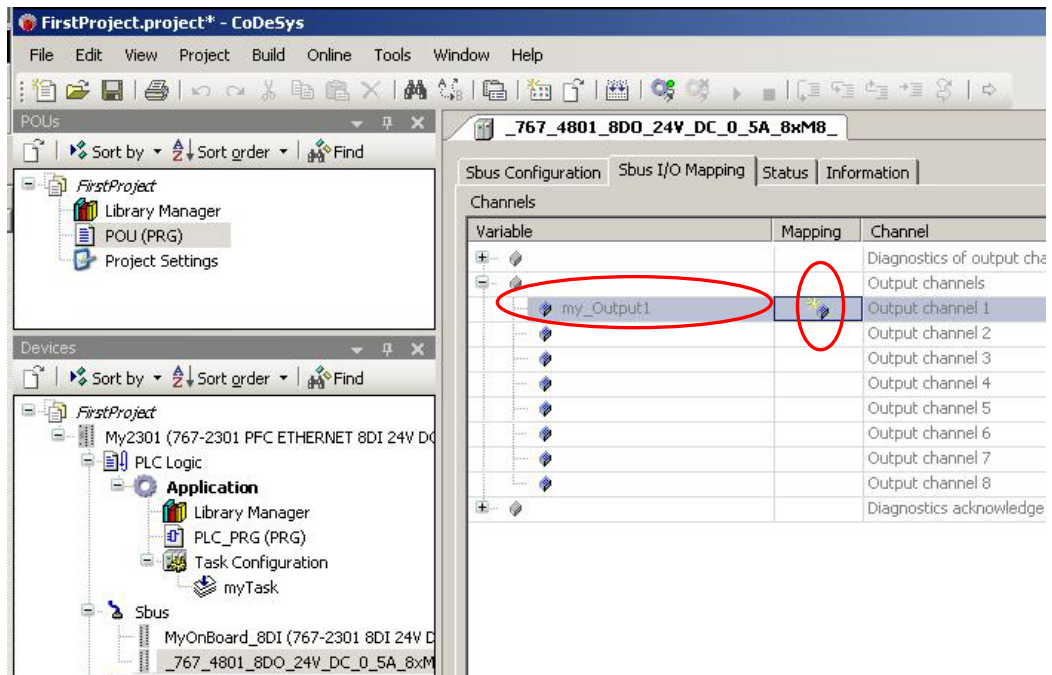


Figure 65: Define variables used in the program using inputs and outputs

3. Enter a name for the data area in the "Variable" pane. By designating the name, you have defined a new variable. This variable can now be used in CoDeSys.

14.3.10 Accessing the MODBUS Register

To access the process data of the MODBUS register, you must add them to the PLC program and assign them a logical MODBUS address.

Both input registers (IR) and output registers (OR) are available:

- **IR**

Read and write access for MODBUS via FC 6
Read access for PLC

- **OR**

Read access for MODBUS via FC 4
Read and write access for PLC



Note

To exchange process data using MODBUS, the MODBUS TCP or UDP box must be selected under "Port" in the Web-Based Management. For more information, see Section 10.2.3. To exchange process data via MODBUS with CoDeSys, configuration register 23 with UnitID 0xFF (255) must be set to 1. This requires a MODBUS TCP communication with the fieldbus coupler.

To access the MODBUS register, proceed as follows:

1. In the "Devices" window, right-click on "Modbus" and select **Add Device...** from the context menu. The "Add Object" dialog box opens.

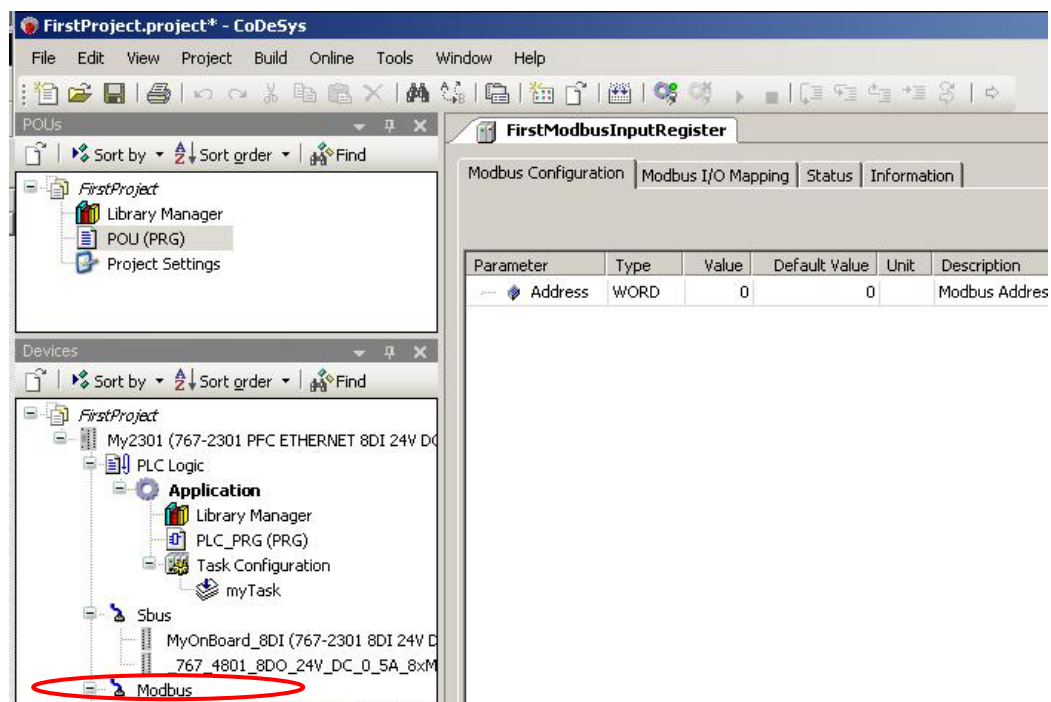


Figure 66: Adding device objects



Figure 67: "Add Device" dialog box

2. Choose here whether you wish to create an input register (MODBUS IR) or an output register (MODBUS OR). Select the corresponding register and click [Add Device]. The selected register appears in the "Devices" window.

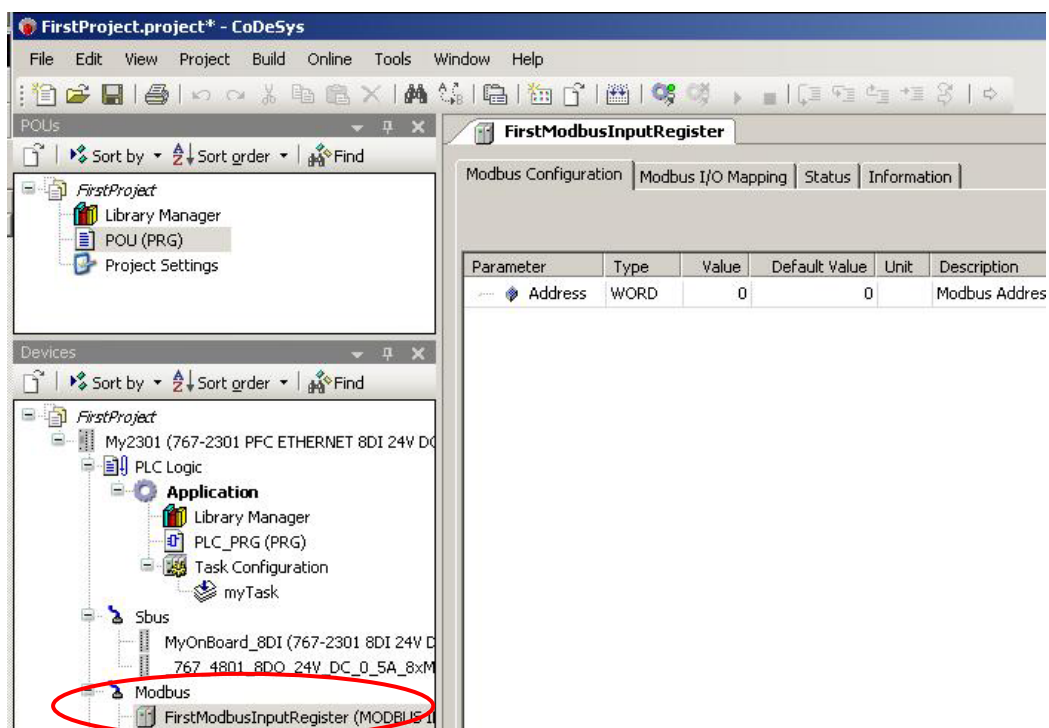


Figure 68: Inserted MODBUS input register

3. To assign a logical MODBUS address to the registers, enter the desired MODBUS address into the "Value" pane. In this example, the MODBUS address is 20.

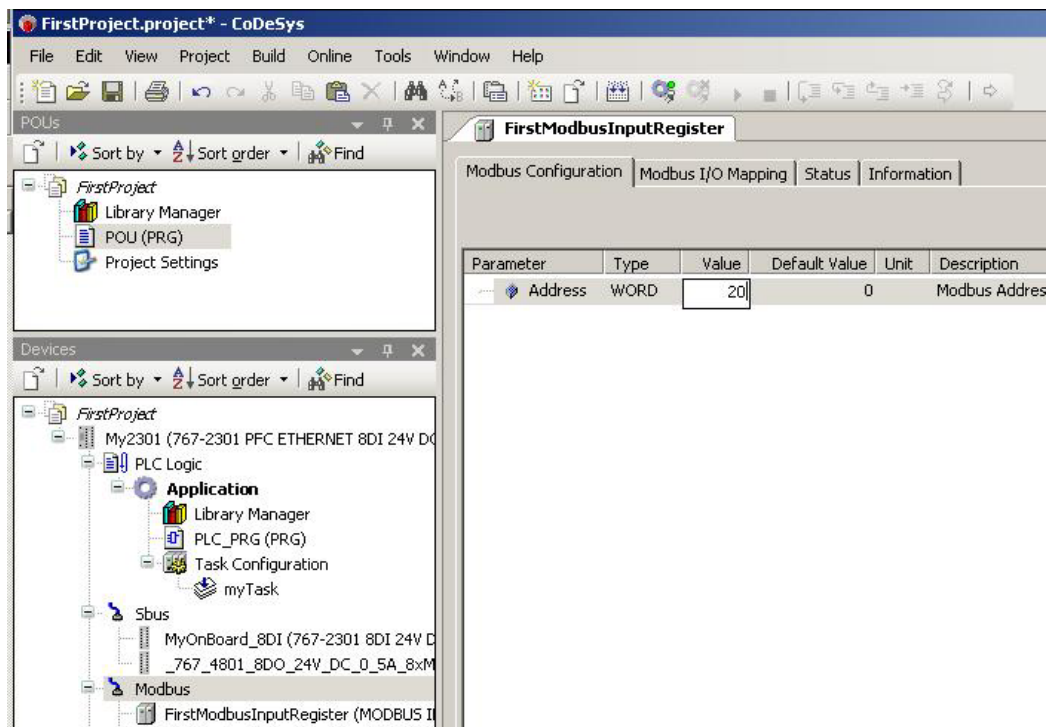


Figure 69: Assigning a MODBUS address

14.3.10.1 Assignment Between MODBUS Registers and the Variables of the Control Program

In order to access the process data of the MODBUS registers in the PLC program, you must connect them with a variable. You have two options:

- Link pre-defined variables with the MODBUS registers
- Define new variables

Linking pre-defined variables with the fieldbus variables

1. To link variables, click on the "Modbus I/O Mapping" tab.
2. Double-click on the icon in the "Mapping" pane to choose between the options "Map to Existing Variable" and "Create New Variable." Select "Map to Existing Variable."
3. In the "Variable" pane, enter the name and path of a variable that you have created in CoDeSys. The following variable is contained in the WAGO project template: Application.PLC_PRG.iCount.

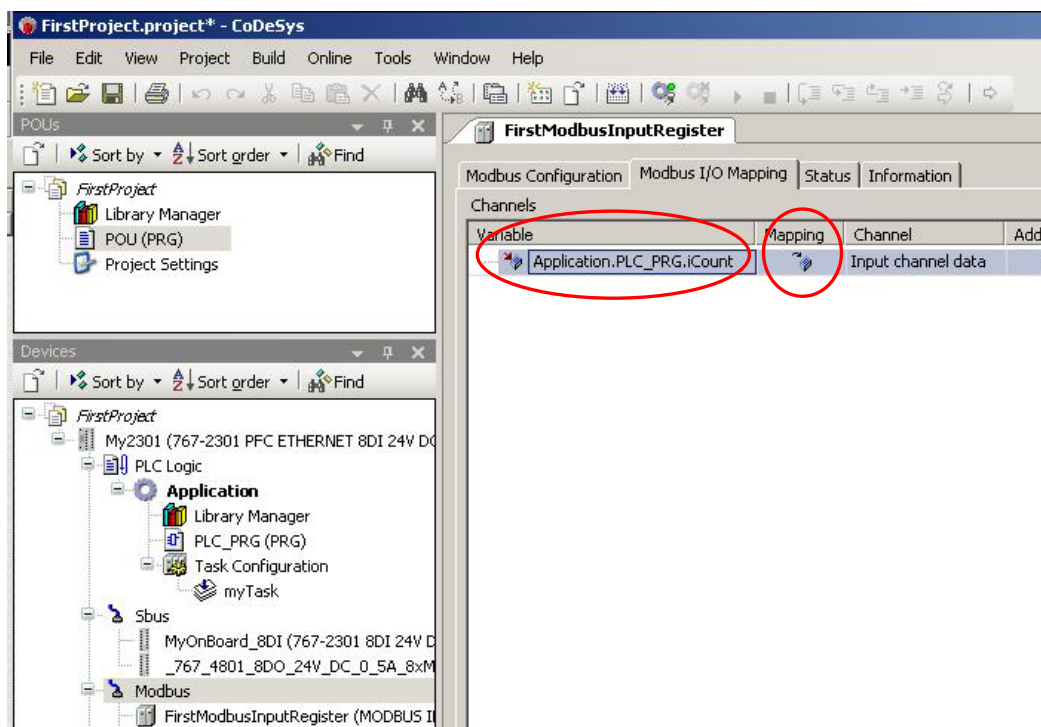


Figure 70: Link variables used in the program using inputs and outputs

Defining New Variables

1. To define variables, click on the "Modbus I/O Mapping" tab.
2. Double-click on the icon in the "Mapping" pane to choose between the options "Map to Existing Variable" and "Create New Variable." Select "Create New Variable."

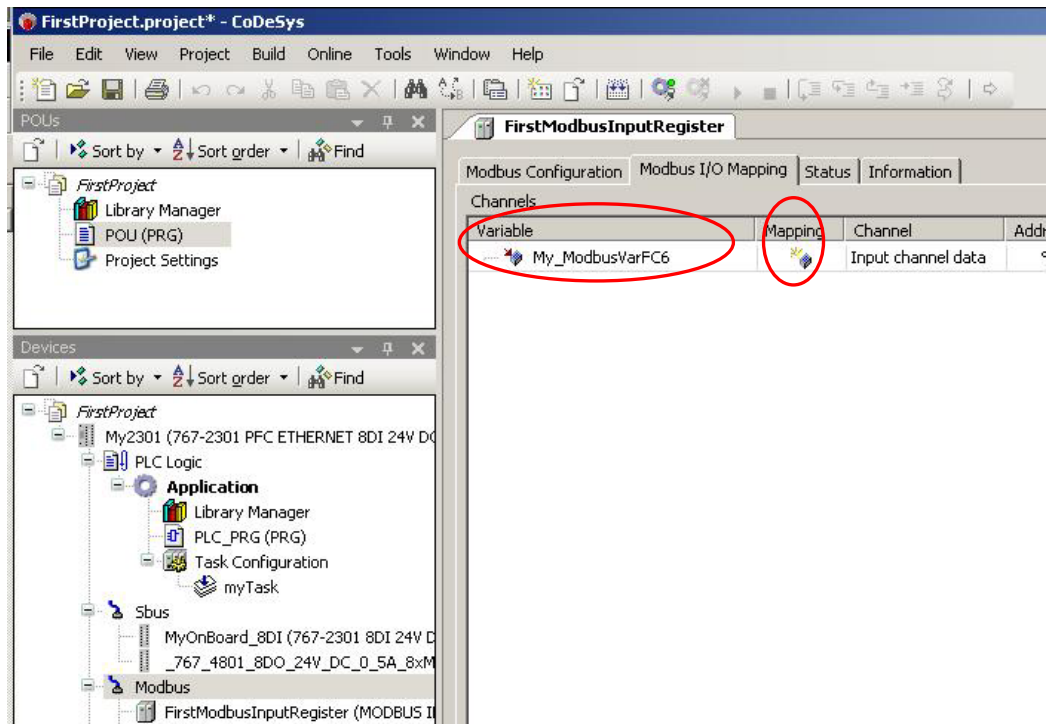


Figure 71: Define variables used in the program using inputs and outputs

3. Enter a name for the data area in the "Variable" pane. By designating the name, you have defined a new variable. This variable can now be used in a CoDeSys program.

14.3.11 Accessing EtherNet/IP Fieldbus Variables

To be able to access the EtherNet/IP fieldbus variables, you must configure the objects used in your application in the PLC program. Each object represents an 8-, 16- or 32-bit value inside the memory area for the input or output variables (per 512 bytes). You can access any address in the memory area using a bit offset. The bit offset is used to determine at which position the EtherNet/IP fieldbus variables are to be read or written in the 512 byte memory area. Corresponding EtherNet/IP objects are assigned to the memory area. These objects are listed in Section 14.3.11.2.

To access the fieldbus variables, proceed as follows:

1. In the "Devices" window, right-click on "Ethernet-IP" and select **Insert Object...** from the context menu. The "Add Device" dialog box opens.

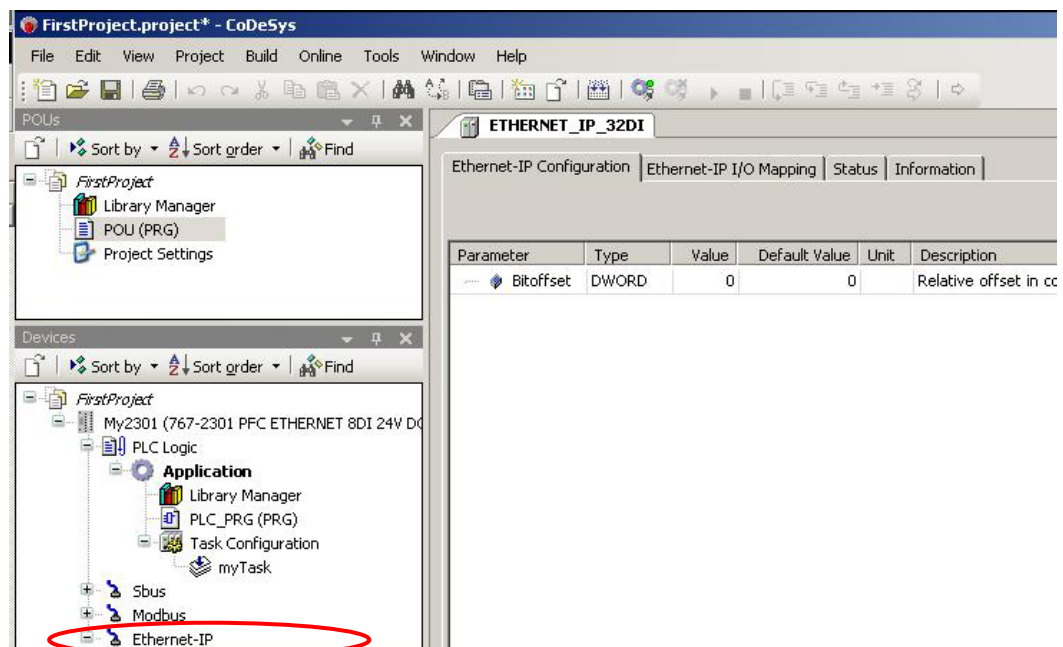


Figure 72: Adding Device

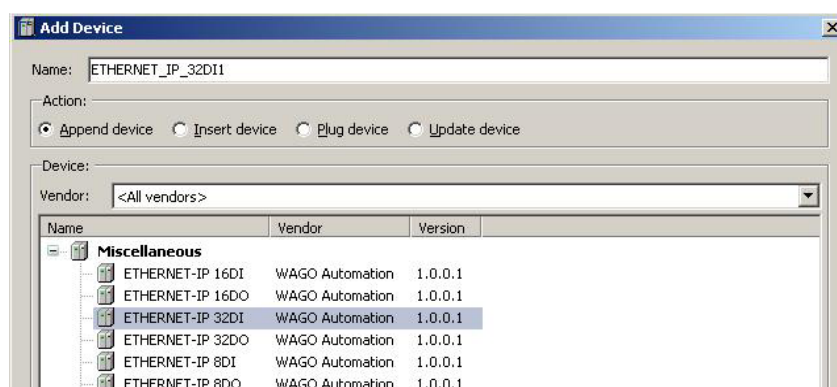


Figure 73: "Add Object" dialog box 1

- In the "Add Device" dialog box, select the desired object according to the size and communication direction of the fieldbus variables used in your application. A "DI" in the object identification refers to a read access to the memory; "a "DO" refers to a write access. You may choose from three different object sizes: 8 bit, 16 bit and 32 bit. Select the appropriate object (e.g., ETHERNET-IP 32DI) and click **[Add Device]**.

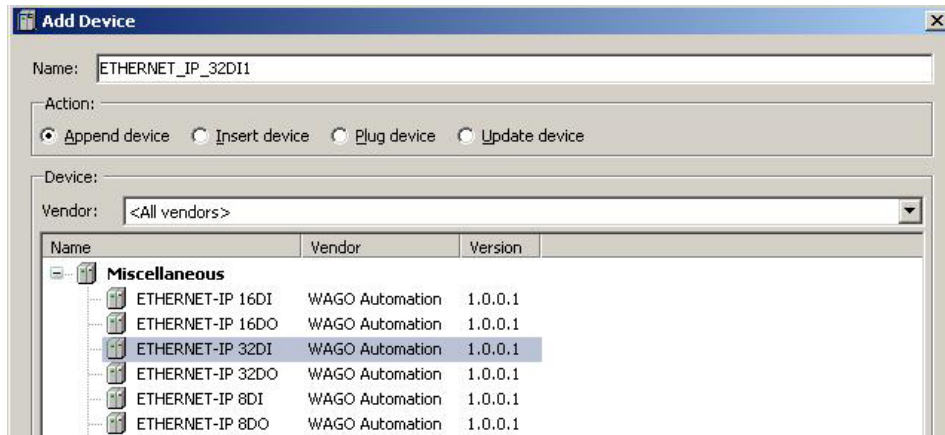


Figure 74: "Add Object" dialog box 2

The desired object is integrated into the "Devices" window.

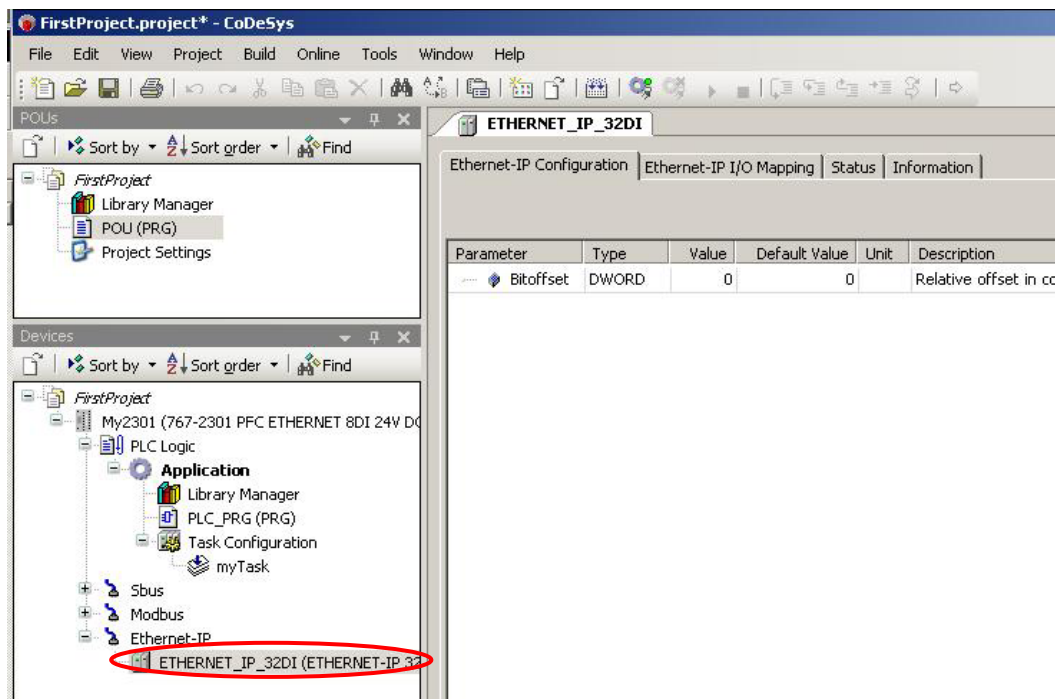


Figure 75: Added EtherNet/IP object

3. Enter bit offset in the "Value" pane of the "Ethernet-IP-Configuration" tab.

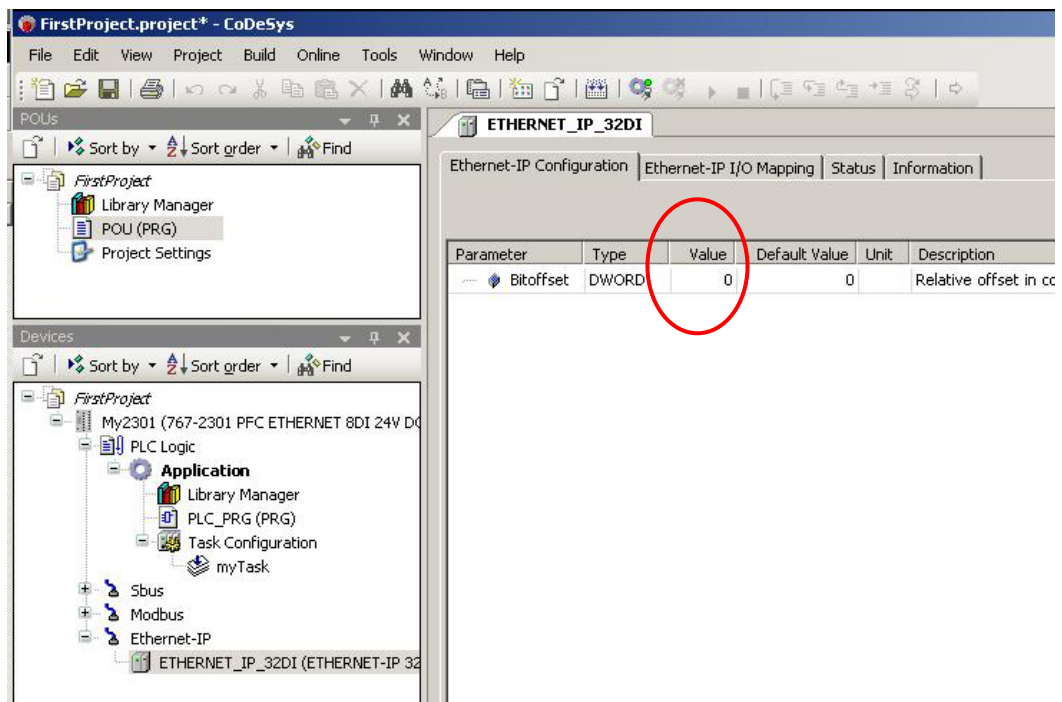


Figure 76: Added EtherNet/IP object with 0 bit offset

Configuration Example for an EtherNet/IP Fieldbus Variable:

You desire to read a 32-bit input value and a 16-bit input value that are completely adjoined. The 32-bit value begins at address 0; the 16-bit value begins at bit offset 32.

To read the 32-bit input value starting from address 0, proceed as follows:

1. Select "ETHERNET-IP 32DI" from the "Add Object" dialog box.
2. Enter bit offset 0 in the "Value" pane. Through this bit offset and the selected 32-bit object (ETHERNET-IP 32DI), this object is restored via EtherNet/IP class AA, instance 1. For more information, see Section 14.3.11.2.

Offset	
0	
8	32DI/AA, 1
16	
24	
32	
40	
48	
...	
511	

3. Assign the EtherNet/IP fieldbus variable to the control program. For more information, see Section 14.3.11.1.

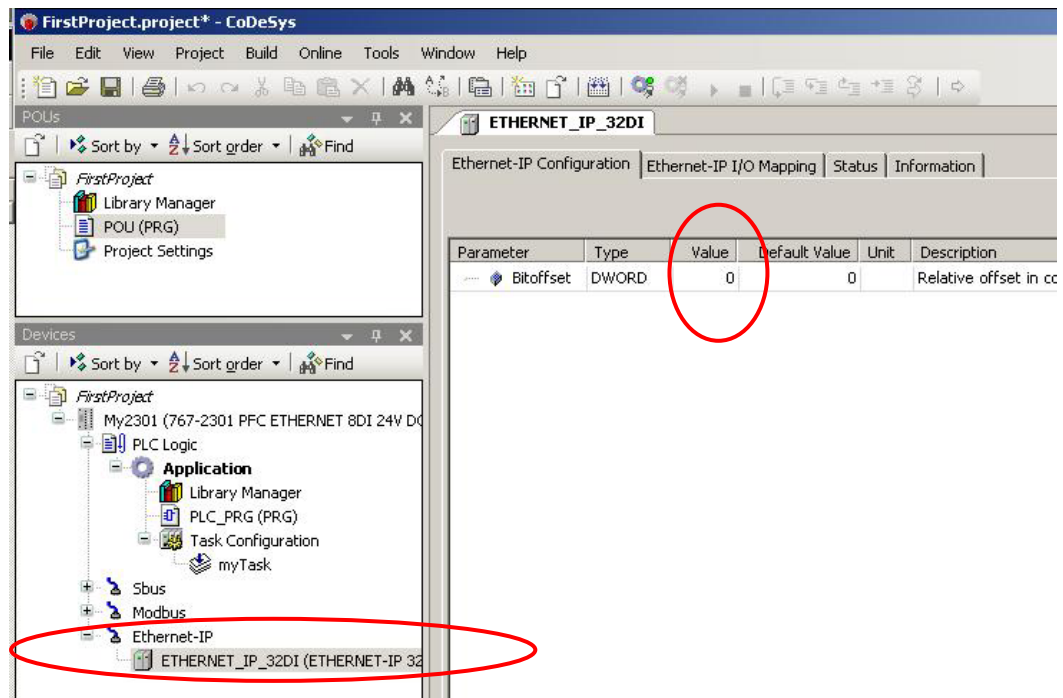
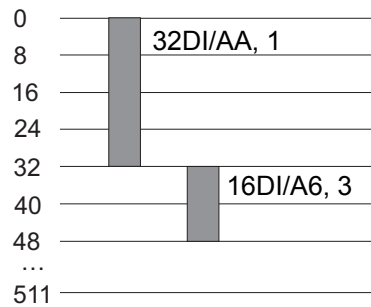


Figure 77: 1. Selected 32-bit object with bit offset of 0

To read the 16-bit input value starting from bit offset 32, proceed as follows:

1. Select "ETHERNET-IP 16DI" from the "Add Object" dialog box.
2. Enter bit offset 32 into the "Value" pane. Through this bit offset and the selected 16-bit object (ETHERNET-IP 16DI), this object is restored via EtherNet/IP class A6, instance 3. For more information, see Section 14.3.11.2.

Offset



3. Assign the EtherNet/IP fieldbus variable to the control program. For more information, see Section 14.3.11.1.

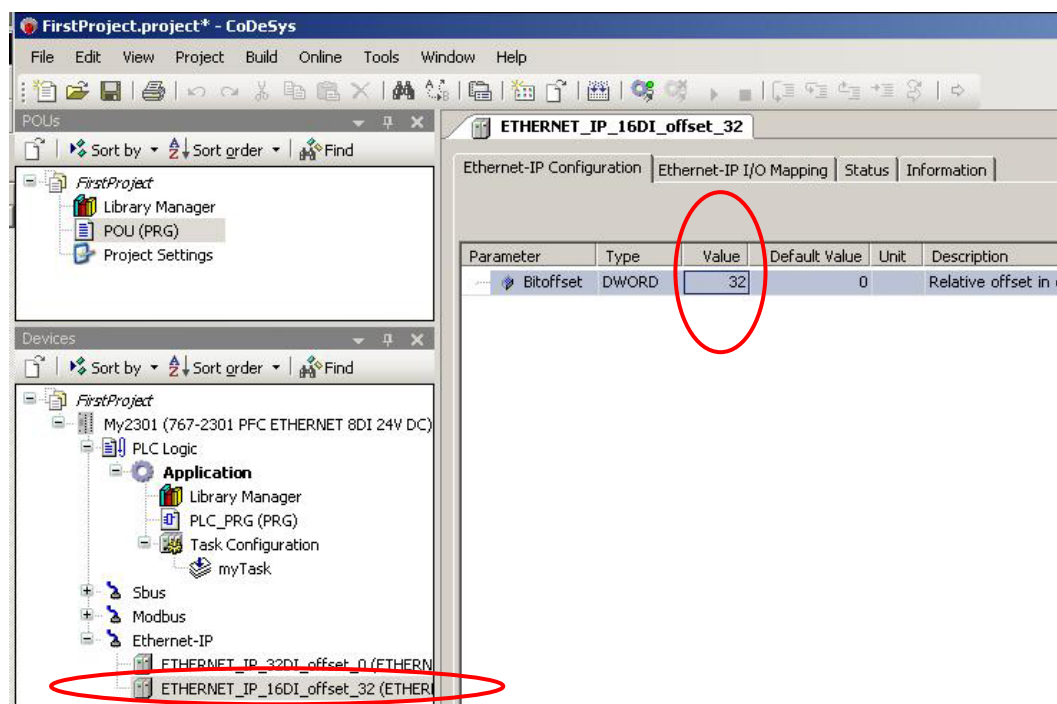


Figure 78: 2. Selected 16-bit object with bit offset of 32

14.3.11.1 Assigning the Fieldbus Variables in the Control Program

To be able to access an EtherNet/IP fieldbus variable, you must connect it with a variable created in the PLC program. You have two options:

- Link pre-defined variables with the fieldbus variables
- Define new variables

Linking pre-defined variables with the fieldbus variables

1. To link variables, click on the "Ethernet-IP I/O Mapping" tab.
2. In the "Variable" pane, enter the name and path of a variable that you have created in CoDeSys. The following variable is contained in the WAGO project template: Application.PLC_PRG.iCount.

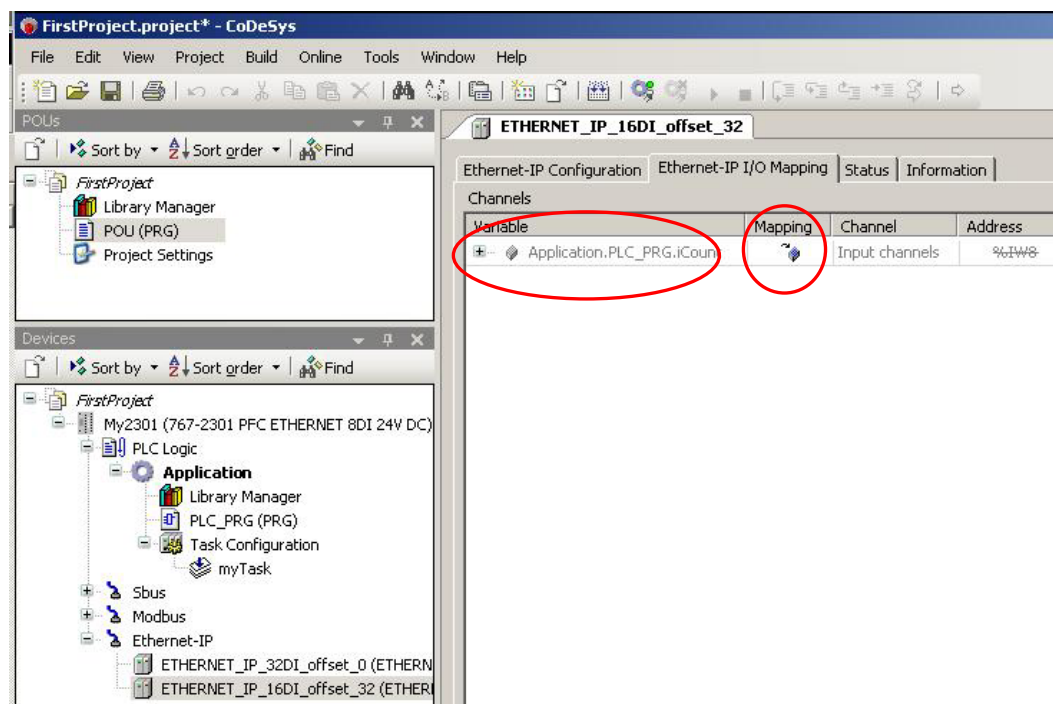


Figure 79: Link variables used in the program using inputs and outputs

3. Adjust the mapping setting. Double-click on the icon in the "Mapping" pane to choose between the options "Map to Existing Variable" and "Create New Variable." Select "Map to Existing Variable."

Icon	Description
	Map to existing variable
	Create new variable

Detailed information on mapping can be found in the CoDeSys online assistance.

Defining New Variables

1. To define variables, click on the "Ethernet-IP I/O Mapping" tab.

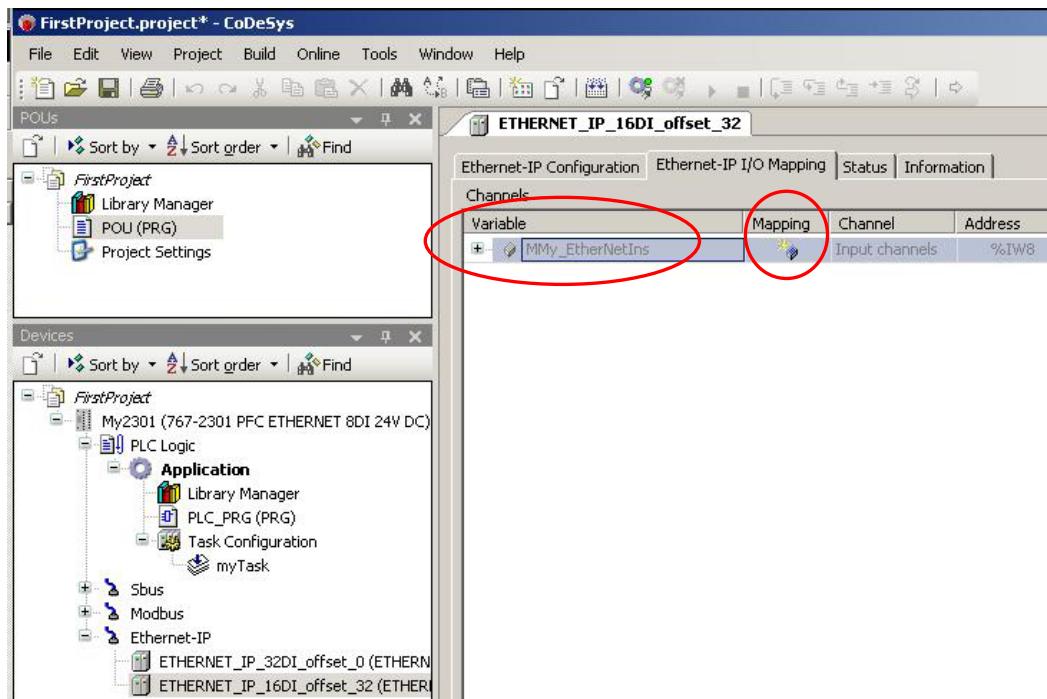


Figure 80: Define variables used in the program using inputs and outputs

2. Enter a name in the "Variable" pane. By designating the name, you have defined a new variable. This variable can now be used in a CoDeSys program.
3. Adjust the mapping setting. Double-click on the icon in the "Mapping" pane to choose between the options "Map to Existing Variable" and "Create New Variable." Select "Create New Variable."

Icon	Description
	Map to existing variable
	Create new variable

Detailed information on mapping can be found in the CoDeSys online assistance.

14.3.11.2 Assigning Fieldbus Variables to the EtherNet/IP Objects

EtherNet/IP Objects of the Fieldbus Input Variables

The values in the "Offset" pane indicate the bit offset in the memory area for the fieldbus input variables (512 bytes); the values in the brackets indicate the byte offset.

Offset	1 byte		1 word		1 double word		1 double word*	
	Class [Hex]	Instance [Hex]	Class [Hex]	Instance [Hex]	Class [Hex]	Instance [Hex]	Class [Hex]	Instance [Hex]
0 (0)	Class A0	Instance 1	Class A6	Instance 1	Class AA	Instance 1		
8 (1)	Class A0	Instance 2						
16 (2)	Class A0	Instance 3						
24 (3)	Class A0	Instance 4						
32 (4)	Class A0	Instance 5	Class A6	Instance 2	Class AA	Instance 2	Class AB	Instance 1
40 (5)	Class A0	Instance 6						
48 (6)	Class A0	Instance 7						
56 (7)	Class A0	Instance 8						
64 (8)	Class A0	Instance 9	Class A6	Instance 3	Class AA	Instance 3	Class AB	Instance 2
72 (9)	Class A0	Instance 10						
...	...	...						
...	...	...						
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
2016 (252)	Class A0	Instance 253	Class A6	Instance 126	Class AA	Instance 63	Class AB	Instance 63
2024 (253)	Class A0	Instance 254						
2032 (254)	Class A0	Instance 255						
2040 (255)	Class A1	Instance 1						
2048 (256)	Class A1	Instance 2	Class A6	Instance 127	Class AA	Instance 64	...	...
2056 (257)	Class A1	Instance 3						
2064 (258)	Class A1	Instance 4						
...	...	...						
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
4064(508)	Class A1	Instance 254	Class A6	Instance 255	Class AA	Instance 128	Class AB	Instance 128
4072(509)	Class A1	Instance 255						
4080(510)	Class A2	Instance 1						
4088 (511)	Class A2	Instance 2						

* Offset of 2 bytes

EtherNet/IP Objects of the Fieldbus Output Variables

The values in the "Offset" pane indicate the bit offset in the memory area for the fieldbus output variables (512 bytes); the values in the brackets indicate the byte offset.

Offset	1 byte		1 word		1 double word		1 double word			
	Class [Hex]	Instance [Hex]	Class [Hex]	Instance [Hex]	Class [Hex]	Instance [Hex]	Class [Hex]	Instance [Hex]		
0 (0)	Class A3	Instance 1	Class A8	Instance 1	Class AC	Instance 1				
8 (1)	Class A3	Instance 2								
16 (2)	Class A3	Instance 3	Class A8	Instance 2						
24 (3)	Class A3	Instance 4								
32 (4)	Class A3	Instance 5	Class A8	Instance 3	Class AC	Instance 2	Class AD	Instance 1		
40 (5)	Class A3	Instance 6								
48 (6)	Class A3	Instance 7	Class A8	Instance 4						
56 (7)	Class A3	Instance 8								
64 (8)	Class A3	Instance 9	Class A8	Instance 5	Class AC	Instance 3	Class AD	Instance 2		
72 (9)	Class A3	Instance 10								
...	...	...	...	...						
...	...	...	...	...					Class AD	Instance 3
...	...	...	...	...	...	...				
...	...	...	...	...	...	...				
...	...	...	...	...	...	...	...	...		
2016 (252)	Class A3	Instance 253	Class A8	Instance 126	Class AC	Instance 63	...	...		
2024 (253)	Class A3	Instance 254					...	...		
2032 (254)	Class A3	Instance 255	Class A8	Instance 127						
2040 (255)	Class A4	Instance 1					Class AD	Instance 63		
2048 (256)	Class A4	Instance 2	Class A8	Instance 128	Class AC	Instance 64				
2056 (257)	Class A4	Instance 3								
2064 (258)	Class A4	Instance 4	...	...					Class AD	Instance 64
...	...	...	...	...			...	...		
...	...	...	...	...	...	...				
...	...	...	...	...	...	...	..	...		
...	...	...	...	...	...	...				
4048 (506)	Class A4	Instance 252	Class A8	Instance 254	...	...				
4056 (507)	Class A4	Instance 253			...	...	Class AD	Instance 127		
4064(508)	Class A4	Instance 254	Class A8	Instance 255	Class AC	Instance 128				
4072(509)	Class A4	Instance 255								
4080(510)	Class A5	Instance 1	Class A9	Instance 1			Class AD	Instance 128		
4088 (511)	Class A5	Instance 2								

14.3.12 Creating a Boot Project and Executing it at Fieldbus Coupler Start

To create a boot project, proceed as follows:

1. Click **Online** in the menu bar and select **Log in to "Application"...**
2. Click **Online** in the menu bar and select **Create boot application for "Application"...**

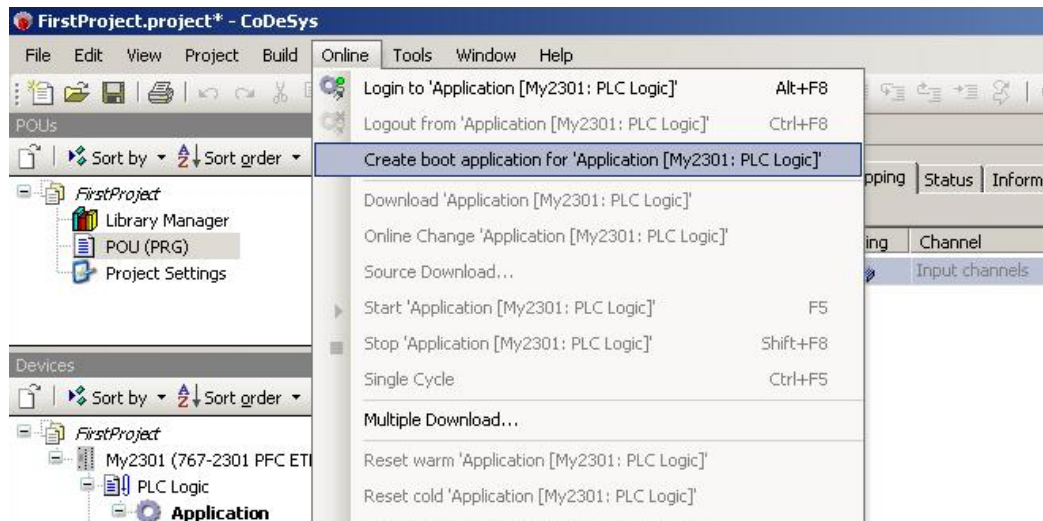


Figure 81: Creating a boot project

1. To monitor whether the boot project has been written into the controller, double-click on your device description in the "Devices" pane. Select the "Files" tab.
2. Click the refresh icon in the "Runtime" window. When the files *Application.app* and *Application.crc* appear, your program is in the controller as a boot project.

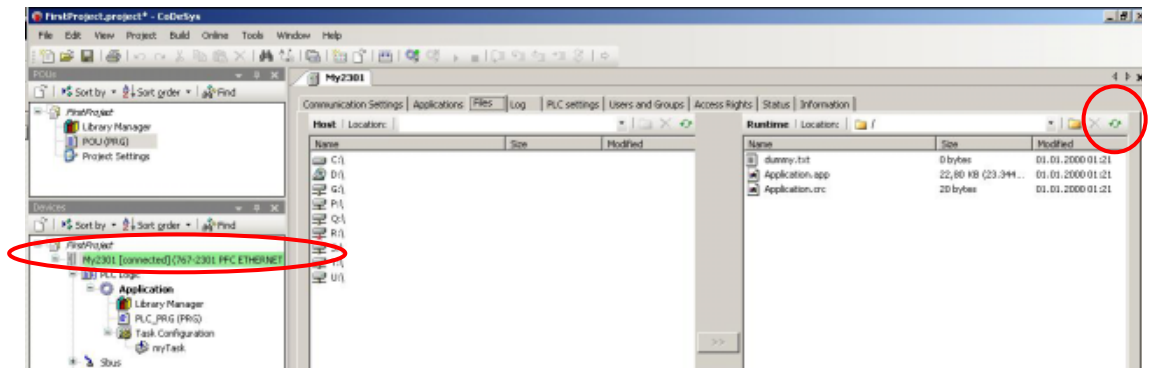


Figure 82: CoDeSys file explorer

Set the switch 10 (Run/Stop) to "ON" (Run) to enable the PLC program to run automatically after each restart.

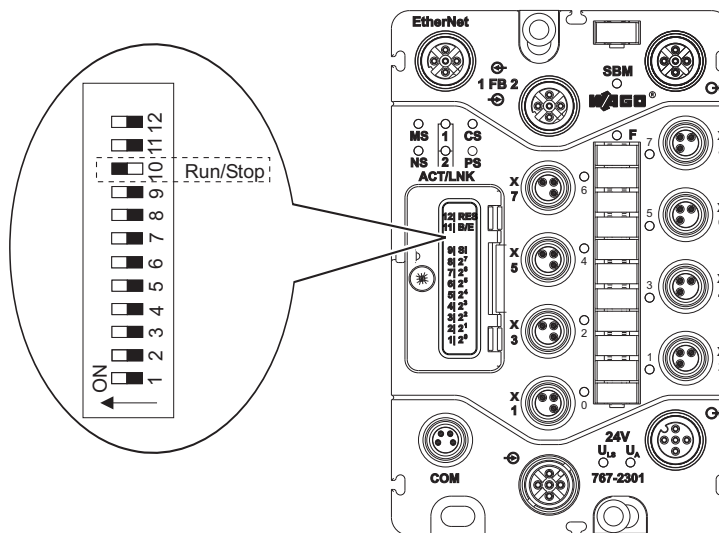


Figure 83: Run/Stop switch (10)

14.4 Notes on CoDeSys Functions

Table 109: Notes on CoDeSys functions

	Description
File system, P drive (1 MB)	The overall size of the PLC program, visualization files, bitmaps, log files, configuration files, etc. may not exceed 1 MB.
Maximum number of IEC tasks	This is limited to 12.
Free-running tasks	Free-running tasks are not supported.
Computer performance/processor time	The fieldbus coupler is based on a real-time operating system with pre-emptive multitasking. High-priority processes such as the PLC program will interrupt or eliminate low-priority processes.
Network load	Because the fieldbus coupler processor also processes network traffic, its load is dependent on it. You can reduce the fieldbus coupler's load by limiting Ethernet communication (e.g., by configuring the integrated switch using WBM). For more information, see Section 10.2.6.
Multi Application	Independent PLC applications can be operated in the fieldbus coupler via "Multi Application" function. Each application has its own memory area and can be changed or loaded independently. For example, this allows lighting control and temperature regulation to be automated independently from each other.

15 Diagnostic and Status Information

For the local diagnostic, the fieldbus coupler contains various LEDs that display the operating state of both the fieldbus coupler and the S-BUS. There is also the possibility for this information to be displayed with WAGOframe. For more information, see Section 15.4.

15.1 Ethernet Status Messages via LED Signals

The following table lists the Ethernet status messages that are signaled by the LEDs (21) on the fieldbus coupler. Information regarding disruption remedies is also provided.

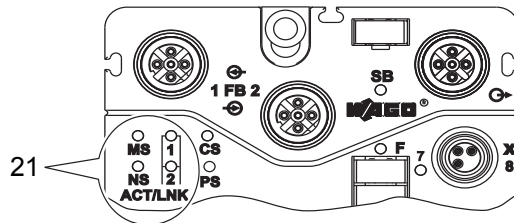


Figure 84: LEDs which display status messages

Table 110: Ethernet status messages

LED	Color/status	Cause	Remedy/information
MS	Off	The fieldbus coupler is not connected to the power supply.	Check the power supply for the fieldbus coupler. Check the cable for damage.
	Green	The fieldbus coupler is operating correctly.	-
	Green flashing	The 767 node is not yet fully configured.	Configure the fieldbus coupler (e.g. by assigning an IP address).
	Red, flashing	A repairable error exists in the 767 node.	Contact WAGO support.
	Red	A non-repairable error exists in the 767 node.	Contact WAGO support.
	Red/green flashing	Self test	-

Table 110: Ethernet status messages

LED	Color/status	Cause	Remedy/information
NS	Off	No IP address has been assigned to the fieldbus coupler. The fieldbus coupler is not connected to the power supply.	Check the power supply for the fieldbus coupler. Check the cable for damage. Assign the fieldbus coupler an IP address.
	Green	A connection to the higher-level controller is established.	-
	Green flashing	No connection to the higher-level controller is established.	-
	Red, flashing	At least one connection has reported a timeout.	Check the "packet rate". Check the cable for damage.
	Red	The IP address used by the fieldbus coupler already exists in the same network.	Assign the fieldbus coupler a different IP address.
	Red/green flashing	Self test	-
ACT/LNK 1	Off	Fieldbus coupler has no connection with the physical network.	Check the cable connection of the FB1 Ethernet connection.
	Green	Connection to physical network exists.	-
	Green flashing	Data is being exchanged via Ethernet.	-
ACT/LNK 2	Off	Fieldbus coupler has no connection with the physical network.	Check the cable connection of the FB2 Ethernet connection.
	Green	Connection to physical network exists.	-
	Green flashing	Data is being exchanged via Ethernet.	-

15.2 Operational Messages of the Fieldbus Coupler via LED Signals

The following table lists the operational messages that are signaled via LEDs. Information regarding remedies of certain causes is also provided.

Note



Use the diagnostic overview (Section 16.7.3) to disable specific diagnostics (see F-LED). In this case, the LED is disabled (off).

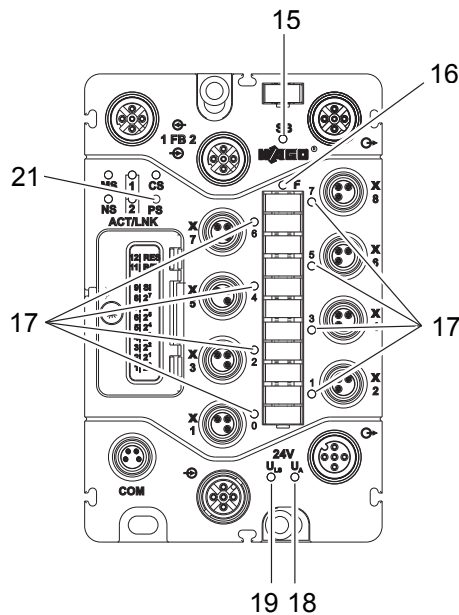


Figure 85: LEDs which display operational messages

Table 111: Operating messages of the fieldbus coupler

Position	LED	Color	Cause	Solution
15	SBM	Green flashing	S-BUS is being started.	-
		Green	S-BUS is functioning without problems.	-
		Red	Disruption on the S-BUS.	Check whether all components are connected to the S-BUS. Also check the S-BUS connection and S-BUS cable
16	F	Red	Group error. At least one diagnostic message is available on the digital inputs.	Check the power supply of the connected sensors.
17	I/O	Yellow	Input signal pending	-

Table 111: Operating messages of the fieldbus coupler

Position	LED	Color	Cause	Solution
18	U _A	Green	Actuator supply is present.	-
		Off	Actuator supply is not present.	Connect the power supply and check the voltage level, if applicable.
19	U _{LS}	Green	Logic supply and sensor supply are present.	-
		Off	Logic supply and sensor supply are not present.	Connect the power supply and check the voltage level, if applicable.
21	PS	Off	There is no PLC program (boot project) in the controller	-
		Green	A PLC program (boot project) is in the controller and is being executed ("RUN" mode)	-
		Red	A PLC program (boot project) is in the controller, but is not being executed ("STOP" mode)	-

15.3 Error messages from the fieldbus coupler via LED signals

Error messages or warnings from the coupler are displayed in WAGOframe. They are also displayed via the CS LED (21) as a blink code.

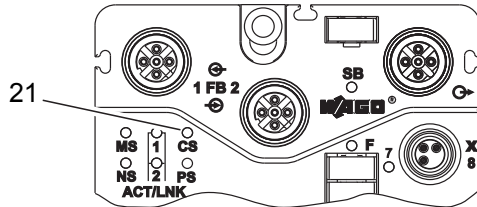


Figure 86: Indicator of blink codes via CS LED

A disruption is always displayed as three blinking sequences in a cyclic manner:

1. The first blinking sequence indicates the **group number**.
The starting point in identifying the error is the group number, from which the error groups can be determined. The number of blink pulses indicates the specific error group.
Example:
Group number 1: Group of S-BUS errors.
2. After a short pause, the second blinking sequence appears and indicates the **error code**. The number of blink pulses indicates the specific error code that describes the type of error.
Example:
Error code 5: S-BUS terminator not attached to last I/O module.
3. After another pause, the third blinking sequence appears and indicates the error argument. The number of blink pulses indicates the **error argument**, which provides supplemental descriptions of the error; for example, on which of the 767 components an error is present.
Example:
Entry 0 ... 63: In which of the connected I/O modules this error has arisen. For example, if "5" is indicated in the blink code, the error arose in the fifth I/O module (the "0" denotes the digital inputs of the fieldbus coupler)

The group number, error code and error argument are displayed as blink codes that are to be converted into numbers. The blink code can display numbers with single, double or triple digits. A zero is always expressed in four cycles of 20 Hz; zeroes preceding a digit(s) are ignored.

15.3.1 Progression of blink sequence

The following table outlines the progression of the blink codes over time. If there is no number in the second position ("1" in the number "10") or third position ("1" in the number "100") for the group number, error code or error argument, the subsequent pause does not take place; instead, the next sequence is initiated (designated in ***bold and cursive*** font in the following table).

Table 112: Overview of the blink codes

Description	Frequency	Additional explanations
Start sequence blinks	10 cycles with 12.5 Hz each (12.5 times per second)	Initiation of start phase
Pause	1 s	
Group number		
Group number (first (100) position)	Repeats: 0.5 s on and 0.5 s off	Repeats according to group number
Pause	2 s	-
Group number (first (10) position)	Repeats: 0.5 s on and 0.5 s off	Repeats according to group number
Pause	2 s	-
Group number (first (1) position)	Repeats: 0.5 s on and 0.5 s off	Repeats according to group number
<i>Pause</i>	<i>2 s</i>	-
<i>blinks</i>	<i>40 ms</i>	Initiation of error code
<i>Pause</i>	<i>2 s</i>	
Error code		
Error code (first (100) position)	Repeats: 0.5 s on and 0.5 s off	Repeats according to error code
Pause	2 s	-
Error code (first (10) position)	Repeats: 0.5 s on and 0.5 s off	Repeats according to error code
Pause	2 s	-
Error code (first (1) position)	Repeats: 0.5 s on and 0.5 s off	Repeats according to error code
<i>Pause</i>	<i>2 s</i>	-
<i>blinks</i>	<i>40 ms</i>	Initiation of error argument
<i>Pause</i>	<i>2 s</i>	
Error argument		
Error argument (third (100) position)	Repeats: 0.5 s on and 0.5 s off	Repeats according to error argument
Pause	2 s	-
Error argument (first (10) position)	Repeats: 0.5 s on and 0.5 s off	Repeats according to error argument
Pause	2 s	-
Error argument (first (1) position)	Repeats: 0.5 s on and 0.5 s off	Repeats according to error argument
Pause	4 s	-
Progression restarts when start sequence blinks		

15.3.2 Example of an error message via blink code

The following example clarifies an error message as indicated by the blink code. A S-BUS error is displayed when the software update for the sixth I/O module fails.

Initialization:

1. The CS LED begins with the initiation of the start phase: quick flashing of about 1s.
2. A one-second pause occurs.

Group number 1: S-BUS error

3. The first position (1) blinks once: 0.5s on and off.
4. The initiation of the error code follows with a pause of two seconds, 40ms of blinking and another pause of two seconds.

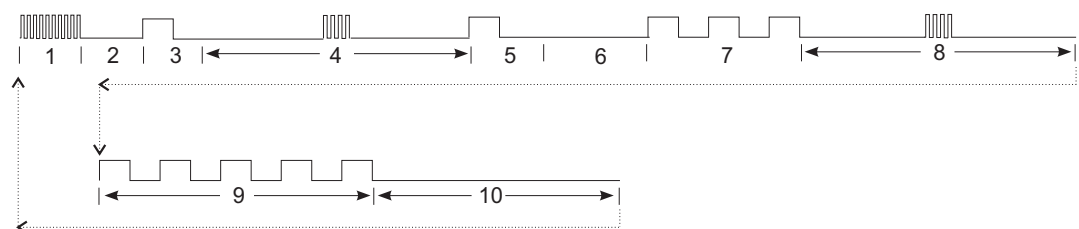
Error code 13: software update for I/O module failed

5. The second position (10) blinks once the 10th position.
6. A two-second pause occurs.
7. The first position (1) blinks 3 times the 1th position.
8. The initiation of the error argument follows with a pause of two seconds, 40ms of blinking and another pause of two seconds.

Error argument 5: I/O module in the sixth slot

9. The first position (1) blinks 5 times the 1th position.
10. A four-second pause occurs.

The blink code starts when the start phase is initiated. This process is repeated if only one error is present; if more are present, the next pending error is processed.



15.3.3 Meaning of the Blink Codes and Procedures for Troubleshooting

In this section, all errors and warnings are listed that are given by the CS LED.

The errors and warnings are divided into the following error groups:

Table 113: List of error groups

Group number	Name	Indicator
1	S-BUS error	CS LED
2	S-BUS warnings	
3	RTS error	
4	-	
5	General internal hardware errors	
6	General internal hardware warnings	
7	General software errors	
8	General software warnings	
9	-	
10	-	
11	Fieldbus-specific software errors	
12	Special software warnings	
13	Firmware loader error	
14	Error with firmware download	

If the subsequent errors and warnings are not corrected with the indicated measures, please contact the WAGO AUTOMATION Support. Be ready to give them the blink code that is displayed.

Tel.: +49 571 887 555
 Fax: +49 571 887 8555
 E-Mail: support@wago.com

Table 114: Group number 1: S-BUS error

Error code	Error argument	Cause	Correction
1	0 ... 65	Error during S-BUS initialization.	Check the cable for damage. Perform a restart by disconnecting the power supply and then reconnecting it.
2			
3			
4			
5	0 ... 64	Error during S-BUS initialization. The last I/O module could not be initialized.	Check the cables to and from the last I/O module. Make sure that the S-BUS terminator is plugged into the last I/O module or are not connected to the fieldbus coupler as 64 I/O modules.
6		Error when starting S-BUS. An S-BUS disruption exists on the I/O module that is located in front of the I/O module indicated by the error argument.	
7	0 ... 63	The "Restart" function (see "S-BUS Master" in Section 16.7.1) is deactivated; a short disruption occurred on the S-BUS. This could not be localized since the disruption on the S-BUS no longer existed during investigation of the disruption point.	Check the cable for damage. Perform a restart by disconnecting the power supply and then reconnecting it.
8			
9			
10			
11	1	A disruption in the S-BUS has arisen.	Check whether the S-BUS cable is connected properly. Check the S-BUS cable for damages.
12	1	Switching to software update mode is not possible.	Perform a restart by disconnecting the power supply and then reconnecting it.
13	0 ... 63	Software update for I/O module failed.	
14	1	Authentication failed.	Perform a restart by disconnecting the power supply and then reconnecting it.
	2		

Table 114: Group number 1: S-BUS error

Error code	Error argument	Cause	Correction
15	0 ... 63	Current software is defective. S-BUS operation only possible in recovery mode to load new software.	Perform a firmware update of the I/O modules.
16	1	Error when starting the S-BUS.	Perform a restart by disconnecting the power supply and then reconnecting it.
	2		
	3	Error when stopping the S-BUS	
17	0	Error at the digital inputs of the fieldbus coupler	Perform a restart by disconnecting the power supply and then reconnecting it.
18	0 ... 63	Communication test with I/O module failed	
19	255	Error when initializing the S-BUS	Perform a restart by disconnecting the power supply and then reconnecting it. If you cannot eliminate the error, please contact WAGO support.
...			
23			
24	1		

Table 115: Group number 2: S-BUS warnings

Error code	Error argument	Cause	Correction
1	1	Cycle time of S-BUS cannot be maintained.	Specify a higher cycle time.
4	255	Firmware update of I/O module	The modules switch to the mode for updating the firmware. In this mode, no data is exchanged (process data).

Table 116: Group number 3: RTS error

Error code	Error argument	Cause	Correction
1	1	Initialization failed when starting the 767 components.	Perform a restart by disconnecting the power supply and then reconnecting it.
1	2	Boot project cannot be loaded.	Reduce the amount of data in the file system. Diminish the project size.
1	4	Checksum not ok. Error in retention memory or boot project.	Reload the boot project to the controller via CoDeSys.
1	5	A reference is used in the PLC program that does not exist in the target.	Use the correct device description file. Delete the parts of the program that contain the unresolved reference.
1	6	Communication error (serial, USB, Ethernet)	Perform a restart by disconnecting the power supply and then reconnecting it. Check the data cables for correct fit and for damages.
1	8	Configuration error	Contact WAGO support.
1	9	Incorrect network mask or incorrect port is being used.	Check the network settings.
1	10	System resources are depleted.	Perform a restart by disconnecting the power supply and then reconnecting it.
1	11	Configuration of PLC application exists twice or is incorrect.	Check the project in use.
1	13	Error of SPEEDWAY driver.	Check the log file in CoDeSys for additional information on the error.
1	14	Error of I/O driver.	Check the project in use.

Table 116: Group number 3: RTS error

Error code	Error argument	Cause	Correction
2	1	Error in PLC program	Check the PLC program.
2	2	CoDeSys has identified an overload or the watchdog time has been exceeded.	Check the timeouts used in the project. Reduce the amount of PLC code.
2	3	The project being used is not appropriate to the device.	Compile the project for the device being used.
2	4	Communication disruption	Perform a restart by disconnecting the power supply and then reconnecting it.
2	5	Error in the scheduler	

Table : Group number 5: general internal hardware errors

Error code	Error argument	Cause	Correction
1	1	Error when accessing flash memory.	Perform a restart by disconnecting the power supply and then reconnecting it. If the problem persists, contact WAGO support.
	2		
2	1	EEPROM error	
	2		
	3	Real-time clock (RTC) is defective	Contact WAGO support.
	4		
3	1	RAM error	Perform a restart by disconnecting the power supply and then reconnecting it. If the problem persists, contact WAGO support.
	2		
	3		
4	1	Error in internal coprocessor	
	2		
5	1		
	2		
6	1	Combination of hardware and software not allowed.	Perform a firmware update on the fieldbus coupler.
7	255	Internal fault	These blink codes assist the support team in further error investigation. Please be ready to give them the group number, error code and error argument.
...			
18			

Table 117: Group number 6: general internal hardware warnings

Error code	Error argument	Cause	Correction
1	1	The fieldbus coupler has not been connected to power supply for six days (back-up capacitor of RTC empty).	Set the real-time clock via WAGOframe.

Table 118: Group number 7: general software errors

Error code	Error argument	Cause	Correction
1	0 – 254	Internal fault	Perform a restart by disconnecting the power supply and then reconnecting it. If you cannot eliminate the error, please contact WAGO support.
2	1 – 7		
3	1 – 4		
4	1		
6	1 – 10		
7	1		
9	1	Internal fault	Perform a restart by disconnecting the power supply and then reconnecting it. If you cannot eliminate the error, please contact WAGO support.
	2		
	3		
	4		
	5		
	10	Error in the parameter database of the fieldbus coupler.	Select the checkbox "Create nominal system configuration" in the SPH-DTM to save the current parameters of the I/O modules again to the file system of the fieldbus coupler (parameter database is created). If you cannot eliminate the error, please contact WAGO support.
	11	Parameter database version is not supported.	
	12	The number of I/O modules does not match the number entered in the parameter database (I/O module removed or added).	
	13	Error in the parameter database of the fieldbus coupler.	
	14	Internal fault	Perform a restart by disconnecting the power supply and then reconnecting it. If you cannot eliminate the error, please contact WAGO support.
	15		
	16		
	20	Order number(s) for I/O module(s) differ(s) from that saved in the parameter database (e.g., 767-6401 instead of 767-6402).	Replace the I/O module by the I/O module with the correct order number. The "Bus address of the first faulty component" parameter appears in the SPH-DTM. This shows you the I/O module to be replaced (e.g., "Bus address 3" refers to the 3rd I/O module on the fieldbus coupler).
9	21	The firmware release index from I/O module(s) does not match that saved in the parameter database.	Update the firmware of the I/O module affected.

Table 118: Group number 7: general software errors

Error code	Error argument	Cause	Correction
	22	The hardware release index from I/O module(s) does not match that saved in the parameter database.	Update the parameter database in the fieldbus coupler for the I/O module affected. Enter the position of the respective I/O module in the "Bus address of I/O modul to update" parameter.
	23	Internal fault	Perform a restart by disconnecting the power supply and then reconnecting it. If you cannot eliminate the error, please contact WAGO support.
	24		
	25	Saved parameters cannot be transferred to the I/O module (communication error).	
	30	Error in the parameter database of the fieldbus coupler.	Select the checkbox "Create nominal system configuration" in the SPH-DTM to save the current parameters of the I/O modules again to the file system of the fieldbus coupler (parameter database is created). If the problem persists, contact WAGO support.
	31	Internal fault	Perform a restart by disconnecting the power supply and then reconnecting it. If you cannot eliminate the error, please contact WAGO support.
	32		
	33		
	34		
	40		

Table 119: Group number 8: general software warnings

Error code	Error argument	Cause	Correction
2	2	Automatic parameterization of the I/O modules is not permitted.	Select the "Do not overwrite I/O module parameters" checkbox in the SPH. Disable this option if you want to allow automatic parameterization for all I/O modules or to update the parameter database.

Table 120: Group number 11: fieldbus-specific software error

Error code	Error argument	Cause	Correction
1	1	Invalid MAC address	Check the MAC address
	2	Assignment of IP address has failed.	Check the configuration of the IP address.
	3	Error during Ethernet initialization.	Perform a restart by disconnecting the power supply and then reconnecting it.
	4		
	4		
	5		
	6		
	7	DHCP lease time elapsed.	Check connection to DHCP server.
	8	IP address already exists in the same network.	Change your IP address.
2	1	Error during FTP initialization.	Contact WAGO support.
	2	Error during SNTp initialization.	
	3	Error during MODBUS initialization.	
	4	Error during EtherNet/IP initialization.	
3	1	Error during SNTp configuration.	Check the configuration of the SNTp parameters.
4	-	Ethernet errors specific to the operating system	Contact WAGO support.
5	1	Fieldbus runtime error	Contact WAGO support.
	2		
	3		
	4		
	5		
	6		

Table 121: Group number 12: fieldbus-specific software warnings

Error code	Error argument	Cause	Correction
5	1	No diagnostic data available.	Perform a restart by disconnecting the power supply and then reconnecting it.
			Check the data cables for correct fit and for damages.
	2	Configured response to a fieldbus error cannot be set.	Perform a restart by disconnecting the power supply and then reconnecting it.
	3		
	4	Internal fault	

Table 122: Group number 13: firmware loader error

Error code	Error argument	Cause	Correction
1	1	No firmware in fieldbus coupler	Perform a firmware update using the USB interface
	2	Firmware checksum error	
	3		
2	1	USB communication disrupted	Perform a restart by disconnecting the power supply and then reconnecting it.
	2		
	3		
	4		
	5		
	6		
	7		
3	1	EEPROM error or incompatible firmware	Perform a restart by disconnecting the power supply and then reconnecting it. Perform a firmware update using the USB interface
	2		
	3		
	4		
	5		
	6		
	7		
4	1	Internal fault	
	2		
5	1		
	2		
6	1	Access to flash memory not possible	
	2		
	3		
	4	Wrong firmware version	
	5	Internal fault	
7	1	Error in firmware file	
	2		
	3		
	4		

Table 122: Group number 13: firmware loader error

Error code	Error argument	Cause	Correction				
	5						
	6						
	7						
8	1	Real-time clock (RTC) is defective	-				
	2						
	3						
9	1	Code processor error	Perform a restart by disconnecting the power supply and then reconnecting it.				
	2						
	3						
	4						
	5						
	6						
	7						
	8						
	9						
10	1			Code processor error	Perform a restart by disconnecting the power supply and then reconnecting it.		
	2						
	3						
	4						
	5						
	6						
	7						
	8						
	9						
11	1	Error when accessing Interrupt Controller				Perform a restart by disconnecting the power supply and then reconnecting it.	
	2						
12	1	SDRAM error					Perform a restart by disconnecting the power supply and then reconnecting it.

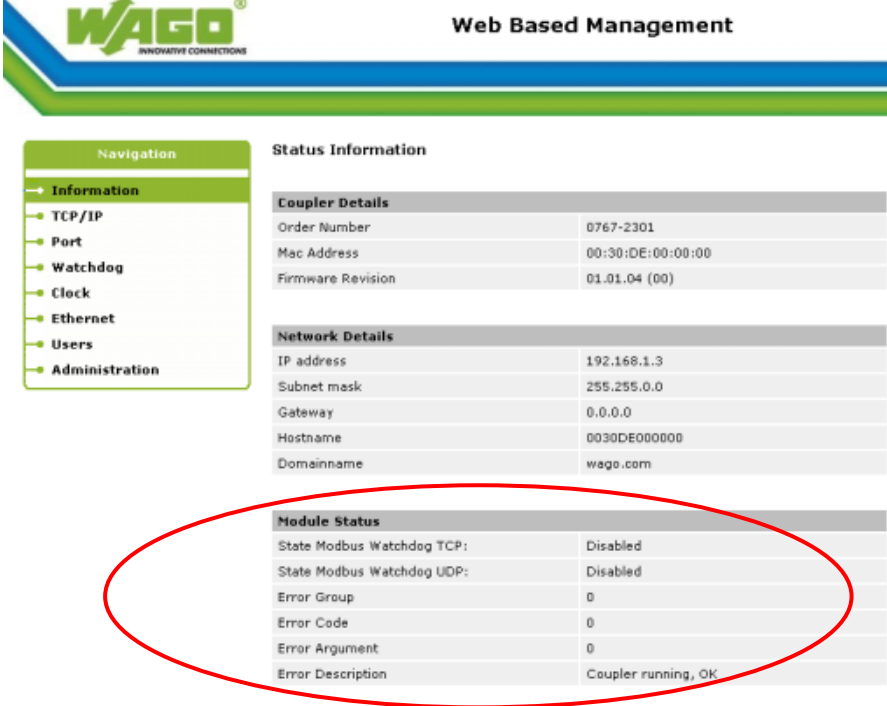
Table 123: Group number 14: error with firmware download

Error code	Error argument	Cause	Correction
1	0	Switching to software update mode is not possible.	Perform a restart by disconnecting the power supply and then reconnecting it.
2	0 ... 64	Software update for I/O module failed.	
3	1	Internal fault	These blink codes assist the support team in further error investigation. Please be ready to give them the group number, error code and error argument.
	2		
	3		
	4		

15.4 Readout of Blink Codes using WAGO DTM

Error messages or warnings from the fieldbus coupler are also displayed in the web-based management on the "Information" page and in WAGOframe under the "Blink code" entry. An overview of the meaning of the blink codes and procedures for correction can be found in Section 15.3.3.

In order to read out the blink codes using WAGOframe, this (or another FDT frame application) and the WAGO DTM must be installed on your PC. For more information, see Section 15.



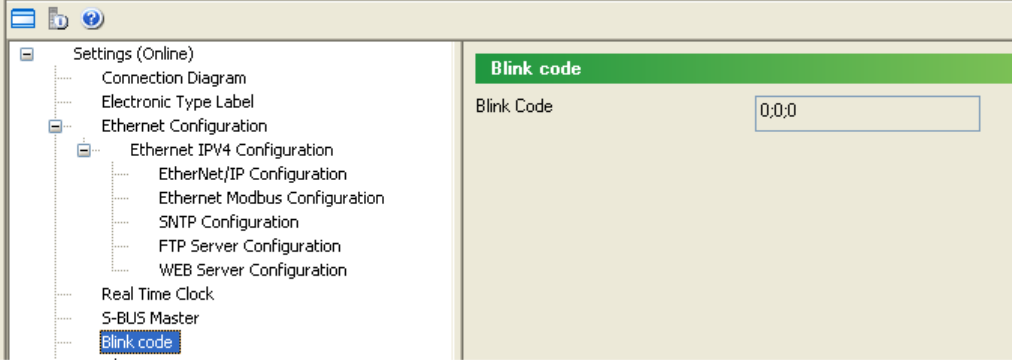
The screenshot shows the WAGO Web Based Management interface. On the left is a navigation menu with options: Information, TCP/IP, Port, Watchdog, Clock, Ethernet, Users, and Administration. The 'Information' section is selected. The main content area is titled 'Status Information' and contains three tables: 'Coupler Details', 'Network Details', and 'Module Status'. The 'Module Status' table is circled in red.

Coupler Details	
Order Number	0767-2301
Mac Address	00:30:DE:00:00:00
Firmware Revision	01.01.04 (00)

Network Details	
IP address	192.168.1.3
Subnet mask	255.255.0.0
Gateway	0.0.0.0
Hostname	0030DE000000
Domainname	wago.com

Module Status	
State Modbus Watchdog TCP:	Disabled
State Modbus Watchdog UDP:	Disabled
Error Group	0
Error Code	0
Error Argument	0
Error Description	Coupler running, OK

Figure 87: Detailed display of the blink code in the web-based management



The screenshot shows the WAGOframe interface. On the left is a tree view of settings: Settings (Online), Connection Diagram, Electronic Type Label, Ethernet Configuration, Ethernet IPv4 Configuration, EtherNet/IP Configuration, Ethernet Modbus Configuration, SNTP Configuration, FTP Server Configuration, WEB Server Configuration, Real Time Clock, S-BUS Master, and Blink code. The 'Blink code' entry is selected. The main area shows a green header 'Blink code' and a text box labeled 'Blink Code' with the value '0;0;0'.

Figure 88: Example of the blink code display under the "Blink code" parameter in WAGOframe

16 Parameterization via FDT/DTM

This Section explains device parameterization with the help of an FDT border application (FDT/DTM) using WAGOframe.

DANGER

Changing parameters!

When parameters are incorrectly modified with an FDT/DTM border application (e.g. WAGOframe), machine components could be placed in a dangerous state and personnel and machines could be at risk. Before changing the parameters, ensure that the machine components are in a safe and defined state and switch off the higher-level controller.

Also ensure before start-up that no personnel remain in the danger area of the machine components.

FDT is an abbreviation for "Field Device Tool". This refers to an application that can be utilized for parameterization of the fieldbus devices independent of the fieldbus being used. For this purpose, the application needs supplementation in the form of software components, which establish the communication with the individual devices and supply the adjustable parameters. These software components are called DTM (Device Type Manager) and produced by the device manufacturers.

FDT/DTM represents an open concept in which the individual components of various manufacturers work together. The concept thus reduces the number of proprietary, manufacturer-specific software solutions and cultivates a uniform operating concept inside a comprehensive operating program.

For the parameterization of a 767 node, an appropriate DTM is available for each 767 component. Use this DTM to parameterize the 767 components. The 767 components can be parameterized either online or offline. The offline mode enables the parameterization of a 767 component that is not yet present. In the offline mode, first store the parameter in a project and later transfer it to the 767 components.

In the online mode, there is a direct connection between the display and the connected 767 components. If a 767 component is in the online mode, its name is displayed in **bold** and *curative* font in the network window.

16.1 Installing the FDT/DTM Components

The following sections deal exclusively with the WAGOframe FDT/DTM frame application.

When using another FDT/DTM frame application, install only the USB driver (item no.: 759-922), the Service Interface DTM (item no.: 759-371) and the WAGO DTM (item no.: 759-361) on your computer. These can be obtained on the Internet at www.wago.com.

Information on configuring the fieldbus coupler using WAGO DTM can be obtained in Section 16.

1. Insert the "WAGOframe" CD into your CD-ROM drive.
2. The start screen appears. Select the desired installation language. A product selection window opens. If Autostart is not activated, open the "Language.htm" file.

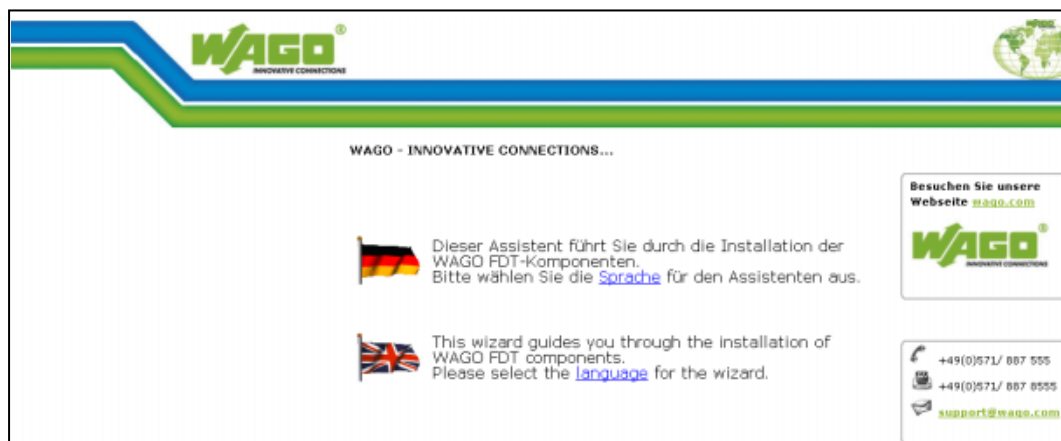


Figure 89: Start screen of WAGOframe CD

3. To install the necessary 767 programs, click on the appropriate link. A window opens displaying the corresponding installation programs.



Figure 90: Window for product selection 1

4. To enable you to use the devices of the 767 series, install the FDT/DTM components shown below by clicking on the appropriate link. The "File Download" dialog box opens. To install the programs, click **[Open]**.
 - USB driver, 759-922 (installation not required when CoDeSys 3 – item no. 759-915 – is installed on your PC)
 - WAGOframe (installation is only necessary if an FDT/DTM border application has been installed on your PC), 759-370
 - WAGO service interface DTM, 759-371
 - DTM for the fieldbus coupler and I/O modules, 759-361
 - DTM for the system update, 759-362



Figure 91: Window for product selection 2

5. If all necessary 767 programs have been installed, connect your PC with the coupler via the USB cable. For more information, please see Section 8.7.

16.2 Starting WAGOframe

1. Start the FDT border application by double-clicking on the WAGOframe logo on your desktop.



Figure 92: WAGOframe logo

2. You can also start WAGOframe from the start menu of your operating system by clicking on "Start" and selecting **Programs > WAGO Software > WAGOframe**.
3. When the "Device Selection Wizard" opens, select "Expert Mode". The "Point to Point Mode" only applies to the configuration of directly connected devices with no S-BUS (such as WAGO Jumpflex). Click [Next].



Figure 93: "Device Selection Wizard"

4. When opened for the first time, the "Query WAGOframe" dialog box appears. Click [Yes] to automatically set up the device catalog on your PC. After the device catalog has been updated, all 767 components for which a DTM is installed are listed.

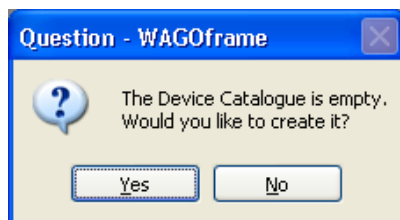


Figure 94: "Query WAGOframe" dialog box

16.3 Expansion of Device Catalog to include 767 Components

WAGOframe automatically detects newly installed DTMs at the next start-up. To expand the device catalog, proceed as follows:

1. To update the device catalog, navigate to the menu bar and select **View > Device Catalog**.
2. In the "Query WAGOframe" dialog box, click **[Yes]** to update the device catalog.

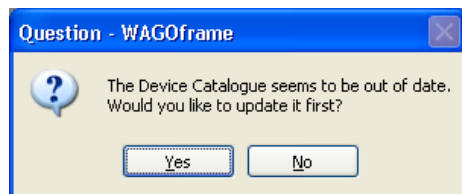


Figure 95: "Query WAGOframe" dialog box

After updating, the device drivers for the 767 components appear in the device catalog.

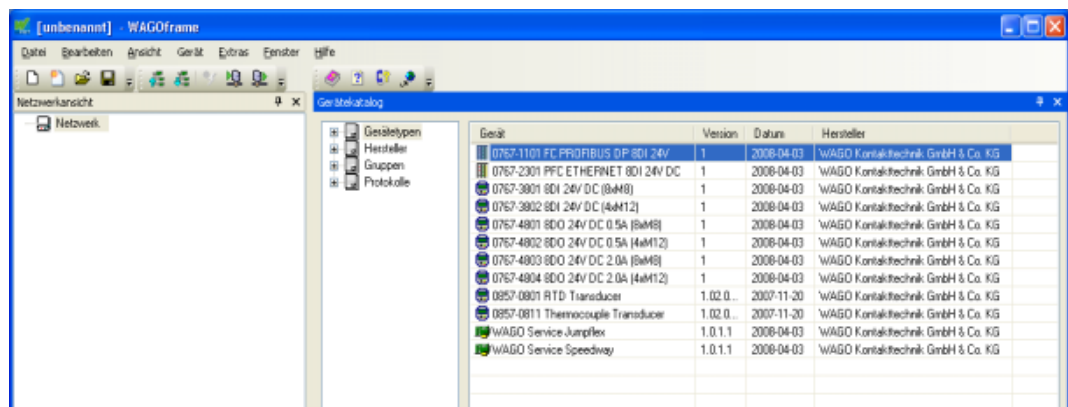


Figure 96: View of device catalog

16.4 Setting Up Network Manually

In order to work with the 767 components, the exact node structure topology must be copied into the "Network View" of WAGOframe.



Note

Depending on the application, you can create the network manually (this Section), automatically (see Section 16.6.7) or using the Lifelist (Section 16.6.8).

16.4.1 Adding the Communication DTM

So that you can parameterize the fieldbus coupler and the I/O modules connected with it, establish a connection to your PC via the USB interface.

1. Right-click on "Network" in the "Network View" pane.
2. In the context menu, select **Add....** The "Add..." dialog box opens.

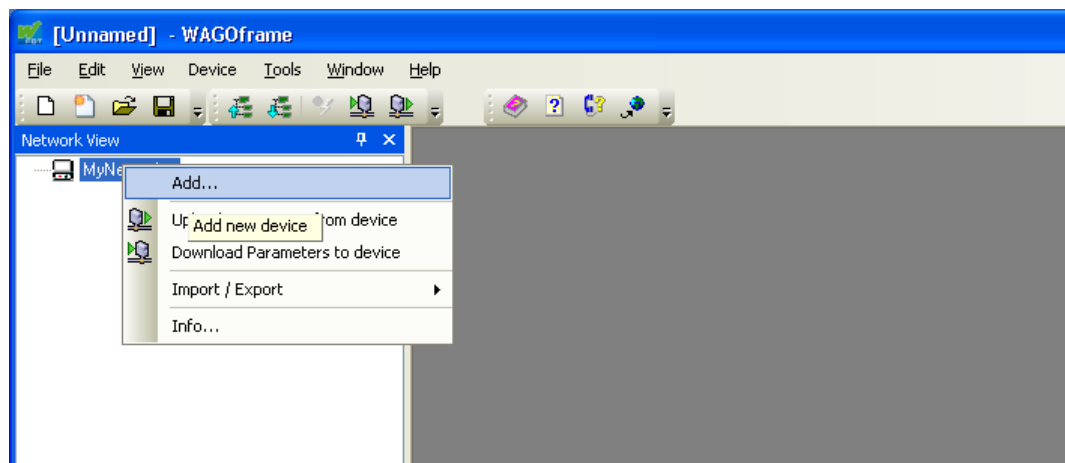


Figure 97: Adding the communication DTM

3. In the "Add" dialog box, select "WAGO Service Speedway".
4. Click **[OK]** to confirm your selection.

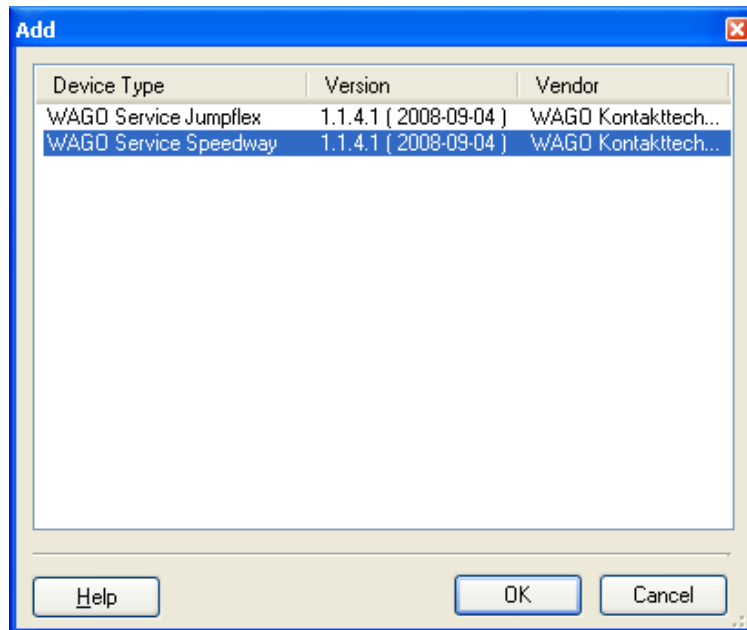


Figure 98: Selecting the communication DTM

16.4.2 Selecting the Communications Interface for WAGOframe

Requirement: The fieldbus coupler must be switched on and your PC must be connected with the USB interface. For more information, see Section 8.7.

1. Double-click on the "WAGO Service Speedway" device driver in the "Network View" pane. The DTM for the interface configuration opens.
2. Select the COM port that you use from the Interface selection box. If the list of available interfaces is empty, check whether the coupler is switched on and connected to your PC via the USB cable.
If you establish the connection via **[Go online]** later, the COM port is checked. If it is incorrect, it is automatically selected.
3. Click **[Apply]** and then **[Close]**.

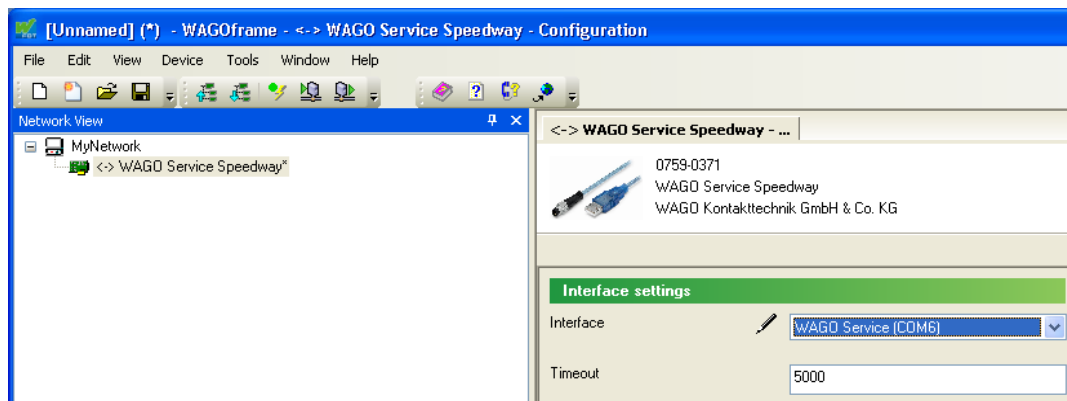


Figure 99: The DTM for the interface configuration



Note

This COM port number remains constant for the fieldbus coupler currently connected. When connecting a different fieldbus coupler to the PC, the COM port also changes. In this case, "WAGO Service Speedway" attempts to select the new COM port automatically. If several fieldbus couplers are connected to the PC, select the correct COM port.

The "WAGO SPEEDWAY Portmapper" is used to specify a constant COM port. The software is contained on the "WAGOframe" CD-ROM.

16.4.3 Adding a Fieldbus Coupler

To incorporate the fieldbus coupler that you use or the programmable coupler into WAGOframe, proceed as follows:

1. Right-click on the "WAGO Service Speedway" device driver in the "Network View" pane.
2. In the context menu, select **Add...**. The "Add" dialog box opens.

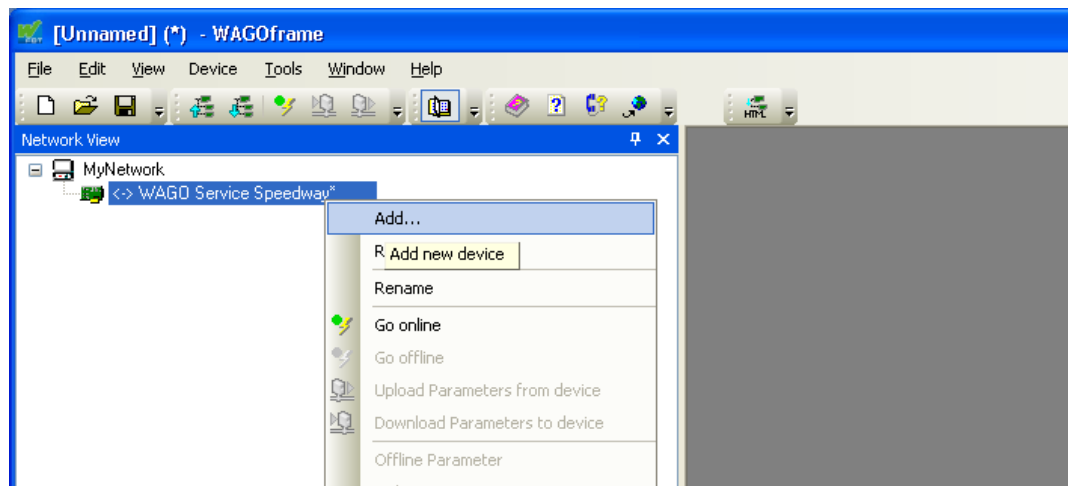


Figure 100: Adding a coupler

3. In the "Add" dialog box, select the coupler that you use.
4. Click **[OK]** to submit your selection.

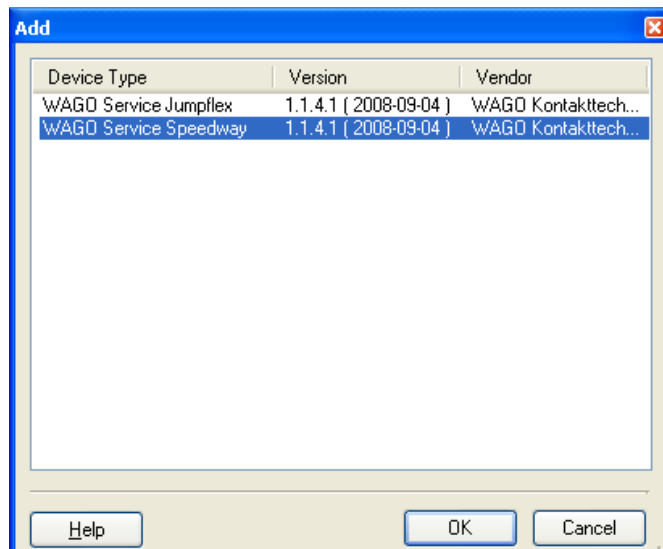


Figure 101: Adding the coupler

16.4.4 Adding the I/O Modules

To incorporate the I/O modules that you use into WAGOframe, proceed as follows:

1. Right-click on the "<Speedway:> 0767-xxxx" device driver in the "Network View" pane.
2. In the context menu, select **Add...**. The "Add" dialog box opens.

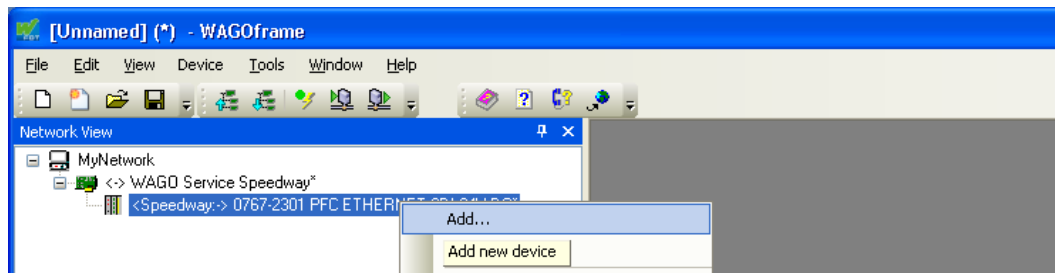


Figure 102: Adding the I/O modules

3. In the "Add" dialog box, select the I/O module's device type.
4. Click [OK] to confirm your selection.

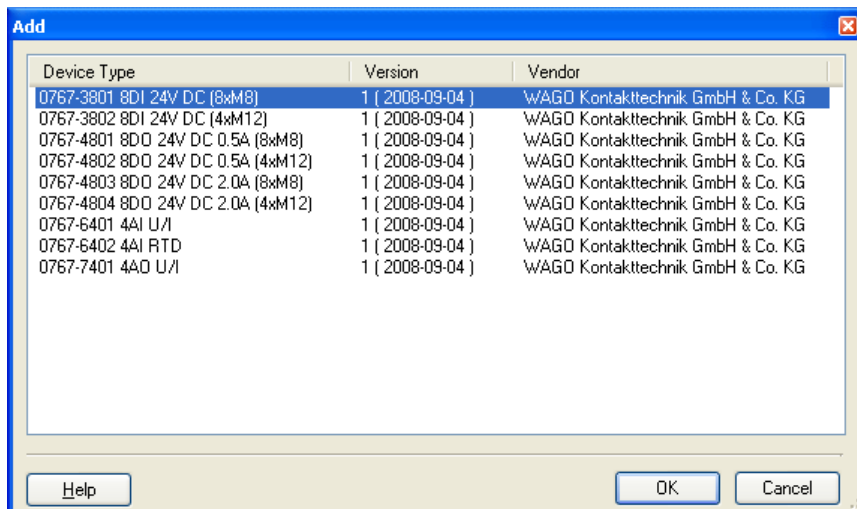


Figure 103: Selecting an I/O module

5. Repeat steps one through four until the module arrangement is concordant with your fieldbus node. In our example, two I/O modules were added.

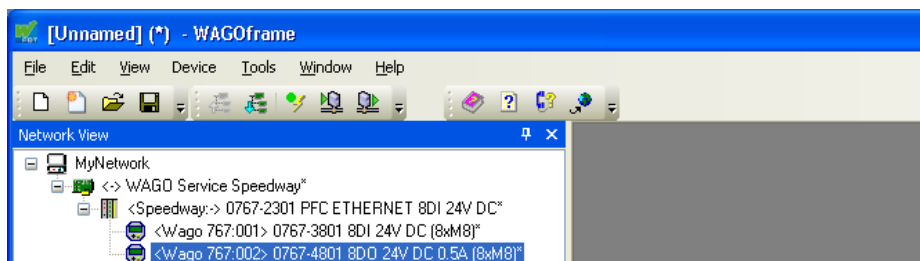


Figure 104: Two added I/O modules

16.5 Online and Offline Configuration

The options of online and offline configuration are available for configuring the 767 components. The offline mode enables the parameterization of a 767 component that is not yet present. In the offline mode, first store the parameters in a project and later transfer them to the 767 components. In the online mode, there is a direct connection between the display and the connected 767 components. If a 767 component is in the online mode, its name is displayed in ***bold and italic*** font in the network window.

16.5.1 Offline Configuration

Requirement:

To configure the fieldbus coupler in offline mode, the network structure of the 767 node must have been transferred to WAGOframe (see Section 16.4).

For offline configuration, proceed as follows:

1. Right-click on the "<Speedway:> 0767-xxxx" device driver of the fieldbus coupler in the "Network View" pane.
2. In the context menu, select **Offline Parameter**. The configuration interface opens with the fieldbus coupler parameters. Details on these parameters can be found in Section 16.7.

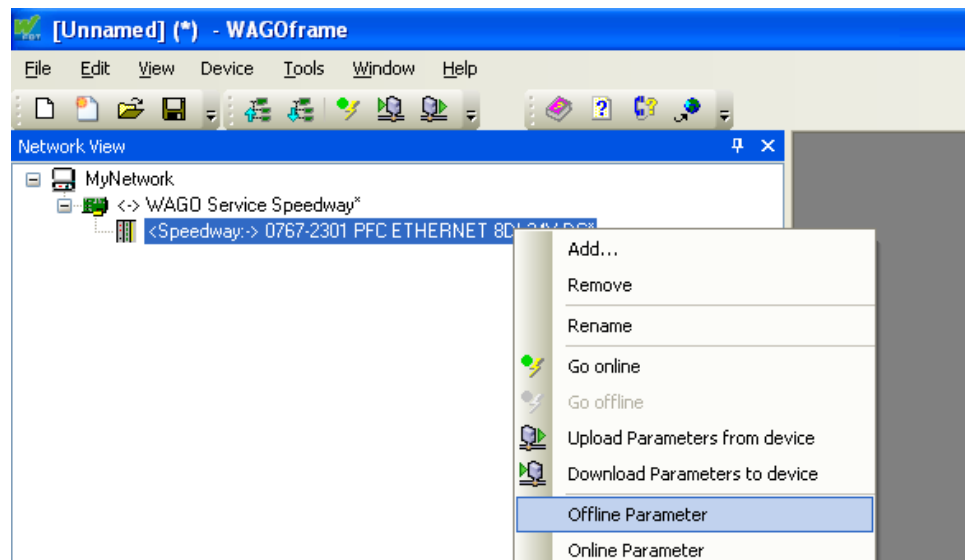


Figure 105: Opening the configuration interface (offline)

3. To configure the I/O modules, right-click on one of the I/O modules in the "Network View."
4. In the context menu, select **Offline Parameter**. The configuration interface opens with the corresponding I/O module parameters. Information on these parameters can be found in the respective I/O module manual.
5. Save the configuration in a project and then transfer it later to the respective 767 component. To transfer parameters, right-click on the fieldbus coupler or on the respective I/O module and select **Download Parameters to device** from the context menu.

16.5.2 Online Configuration

Requirement:

To configure the fieldbus coupler in online mode, the network structure of the 767 node must have been transferred to WAGOframe (see Section 16.4). You can also utilize the "Set Up Network" option as described in Section 16.6.7.

For offline configuration, proceed as follows:

1. Right-click on the "<Speedway:> 0767-xxxx" device driver in the "Network View" pane.
2. In the context menu, select **Go online**. When the progress bar reaches 100%, a connection to the fieldbus coupler has been established. Repeat this step for each desired I/O module.

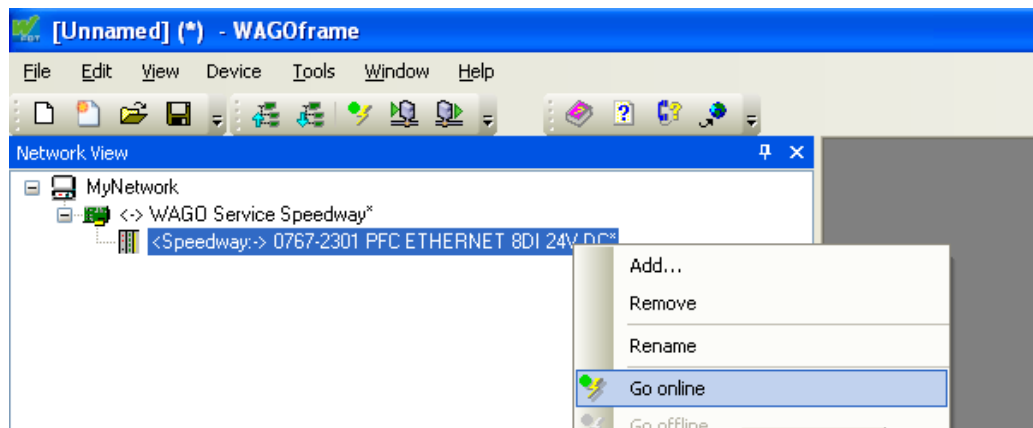


Figure 106: Setting up a connection to the fieldbus coupler

3. In the context menu, select **Online Parameter**. The configuration interface opens with the fieldbus coupler parameters. Details on these parameters can be found in Section 16.7.

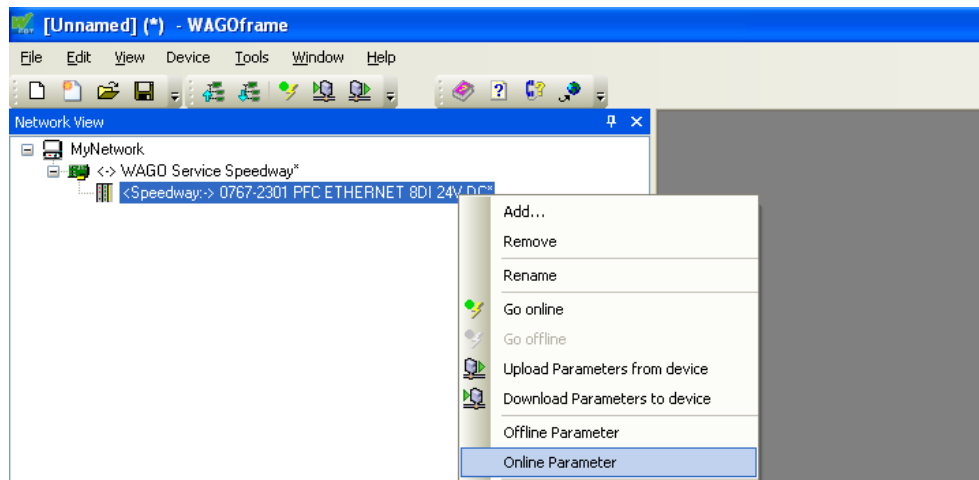


Figure 107: Opening the configuration interface (online)

4. To configure the I/O modules, right-click on one of the I/O modules in the "Network View."
5. In the context menu, select **Online Parameter**. The configuration interface opens with the corresponding I/O module parameters. Information on these parameters can be found in the respective I/O module manual.

16.6 The "Additional Functions" and "Scan" Selections

The "Additional Functions" selection from the context menu provides the following additional functionalities in addition to the possibility of parameterization:

- **Change bus address**
Here you can change the S-BUS addresses for individual I/O modules.
- **I/O owner assignment**
Here you can specify to which process image (e.g., fieldbus or runtime system) an I/O module should belong.
- **Diagnostics setup**
Here you can enable or disable synchronous diagnostics and diagnostic confirmation of the I/O modules, as well as the integrated digital inputs of the fieldbus coupler.
- **Service page**
You can use the Service page to restore the default state for selected 767 components.
- **User administration**
In the user administration, you can change the preset passwords (see Section 11.1) for the **guest**, **user** and **admin** users. In addition, you can reset all passwords to their default state.
- **File system**
Here you administer the file system of the fieldbus coupler.

The "Scan" selection from the context menu offers the following options:

- **Set up network**
Add all 767 components connected to the PC by the USB cable to the network view.
- **Life list**
Here you can search for connected 767 components.

16.6.1 Changing the Bus Address

If you wish to simply configure select I/O modules instead of the entire topology, add these I/O modules manually to the Network View. For these I/O modules to be accessible at the correct bus address, you must assign them the correct addresses. To do so, please proceed as follows:

1. Right-click on the coupler in the "Network View" pane.
2. In the context menu, select **Additional Functions > Change bus addresses**. A window opens with a list of the bus nodes.

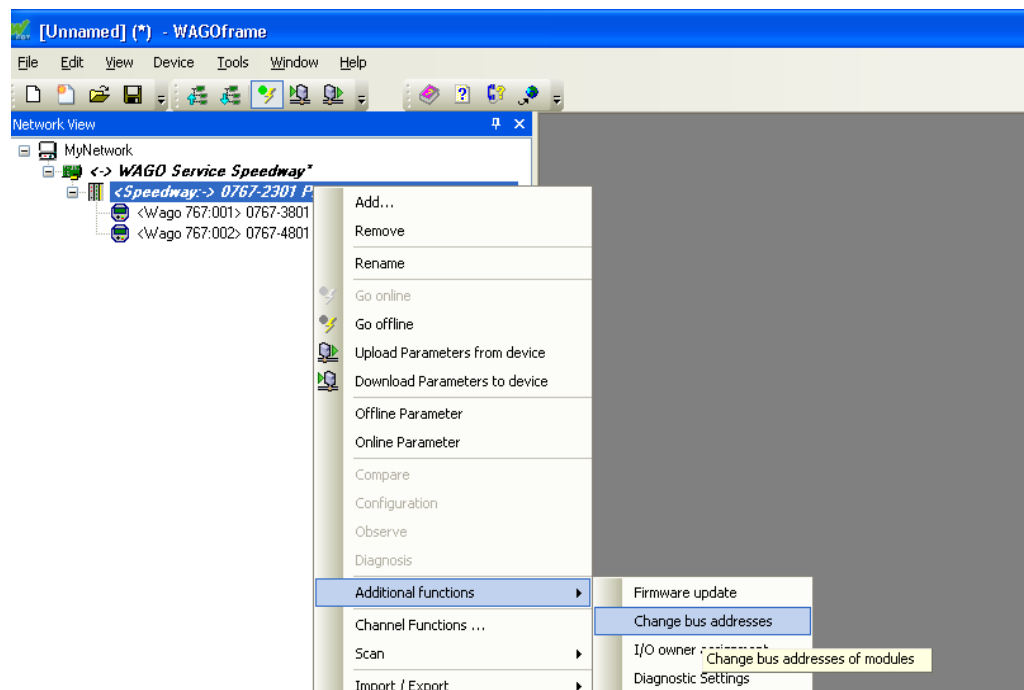


Figure 108: Opening the "List of Bus Nodes" window

- Now select an I/O module from the "List of Bus Nodes" to which you wish to assign a new bus address. The current address is displayed in the **New Bus Address** field. Enter here the desired new address and click **[Apply]**.

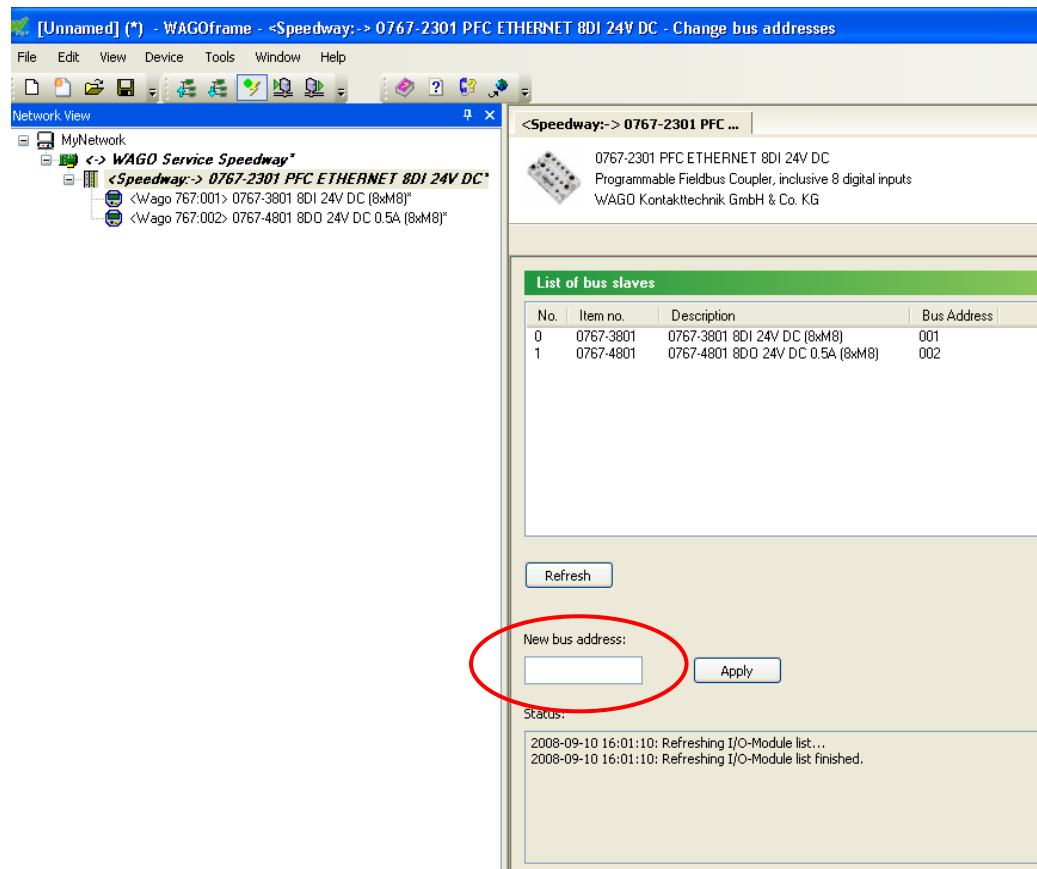


Figure 109: Assigning new bus addresses for the I/O modules

16.6.2 Assigning the I/O Owner

To configure the assignment of an I/O module to a fieldbus or to the runtime environment proceed as follows:

1. Right-click on the coupler in the "Network View" pane. In the context menu, select **Additional Functions** > **I/O owner assignment**. A window opens with a list of the bus nodes.

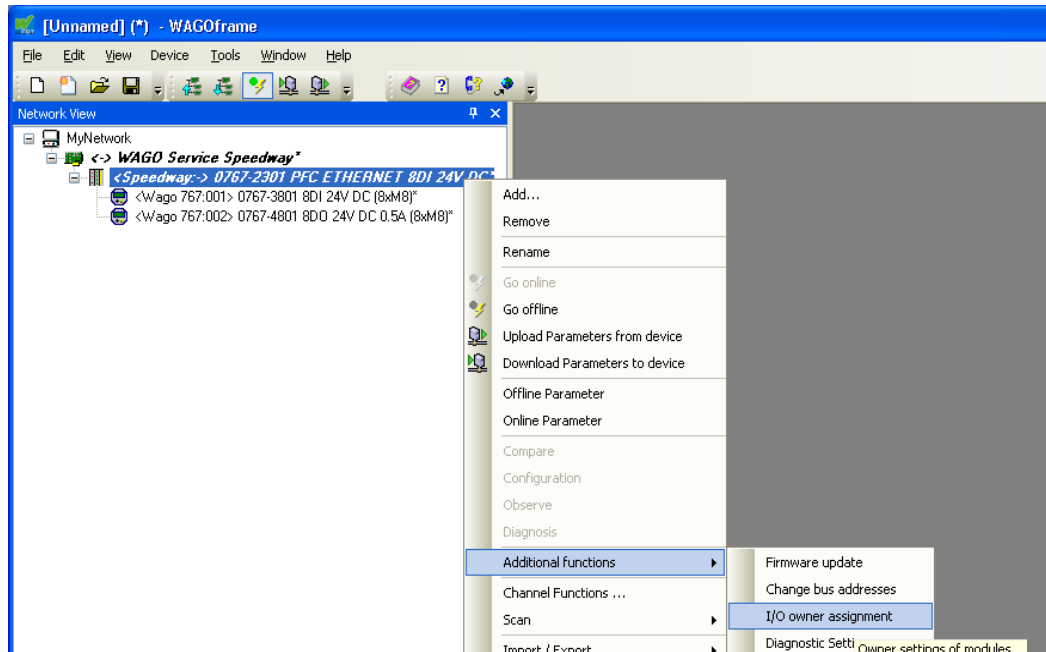


Figure 110: Opening the "Assign Owner" window

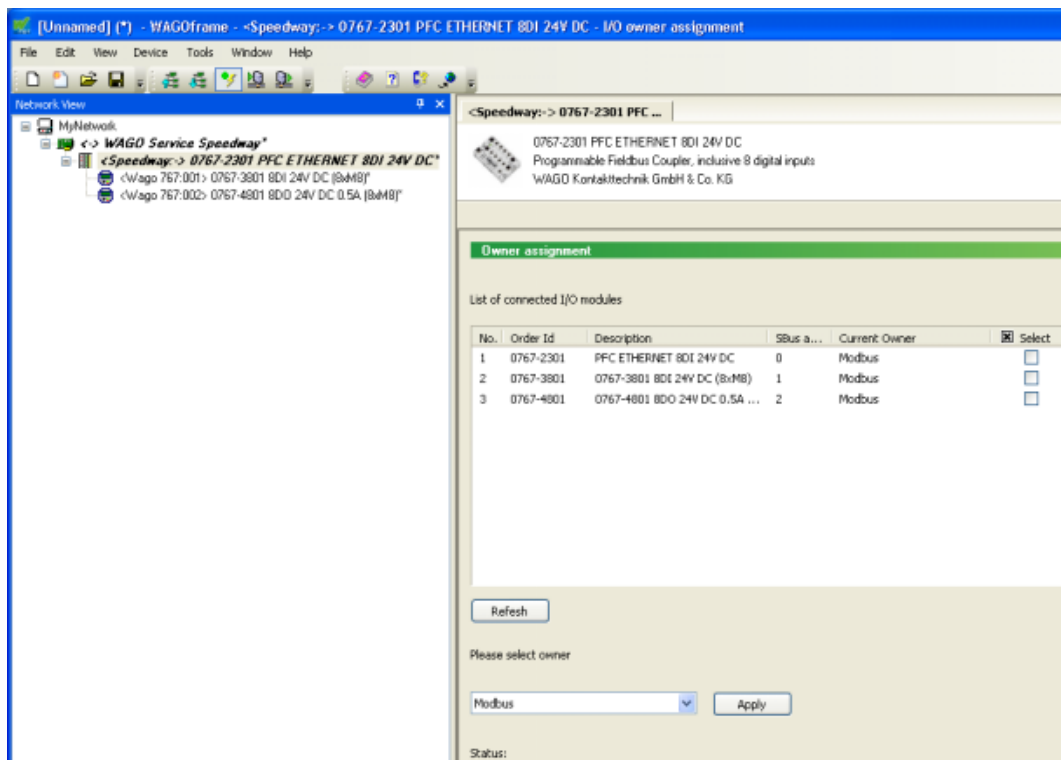


Figure 111: "Assign Owner" window

2. Select the I/O module that you wish to configure via the "Select" selection box.
3. Select the new user (fieldbus or runtime system) for the selected I/O modules from the selection box. Click **[Apply]** to confirm.

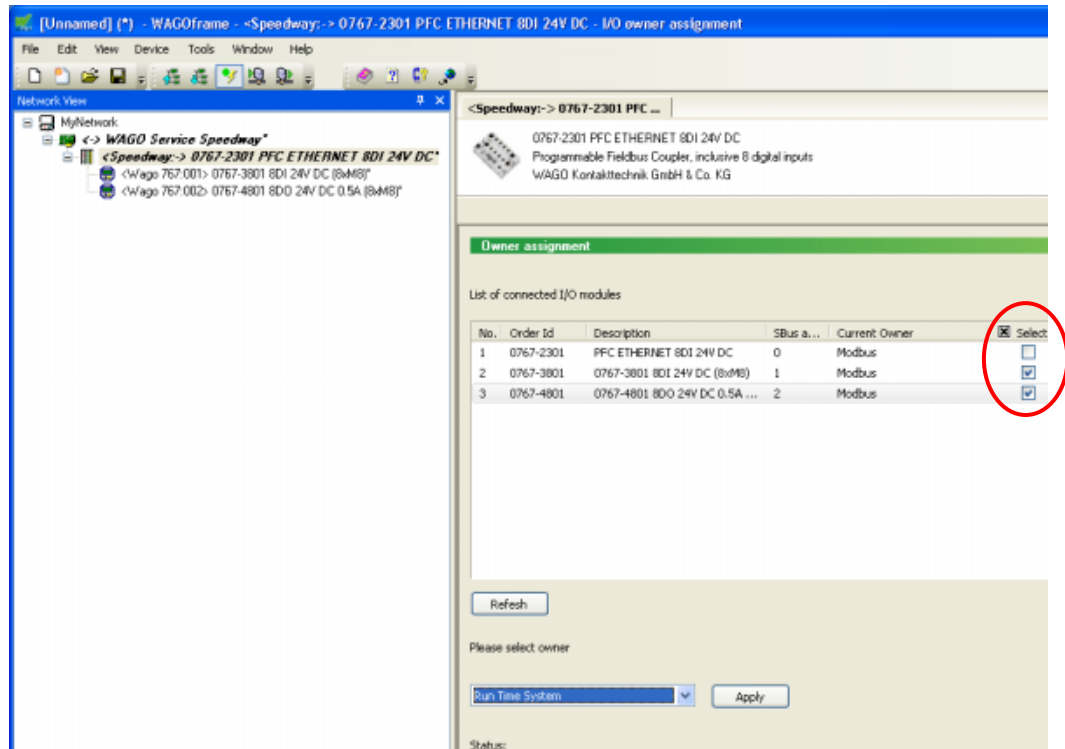


Figure 112: Selecting a new user for the selected I/O modules



Note

If you want to apply the settings, click **[Yes]** in the window. The 767 node is restarted.

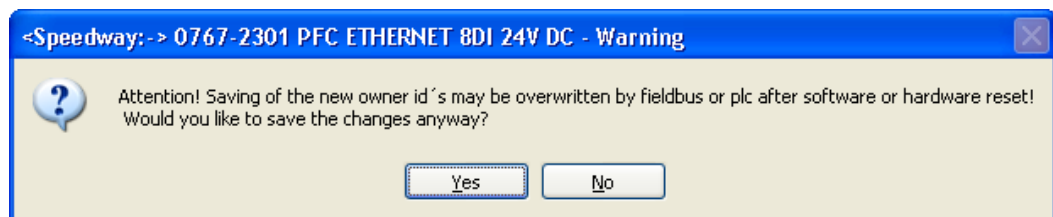


Figure 113: Warning

4. If the changes were successfully carried out, they can be seen in the updated list.

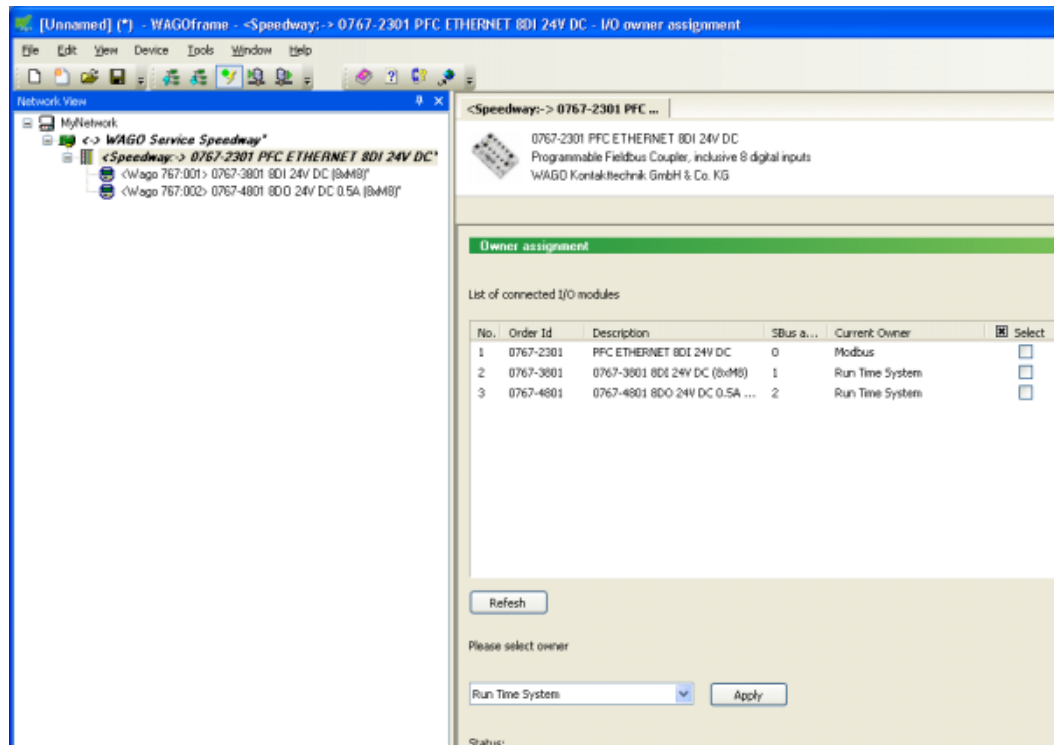


Figure 114: Runtime system chosen as new user for the selected I/O modules

Note



Because configuring the fieldbus or runtime environment accurately sets the assignment, you will normally not need to change the assignments. Configuration is only necessary if the assignment is incorrect.

16.6.3 Diagnostic Setup

Using diagnostic setup, you can activate or deactivate the synchronous diagnostics and the diagnostic confirmation of the I/O modules and couplers.

1. Right-click on the coupler in the "Network View" pane. In the context menu, select **Additional Functions > Diagnostic Settings**. A window opens with a list of the connected 767 components.

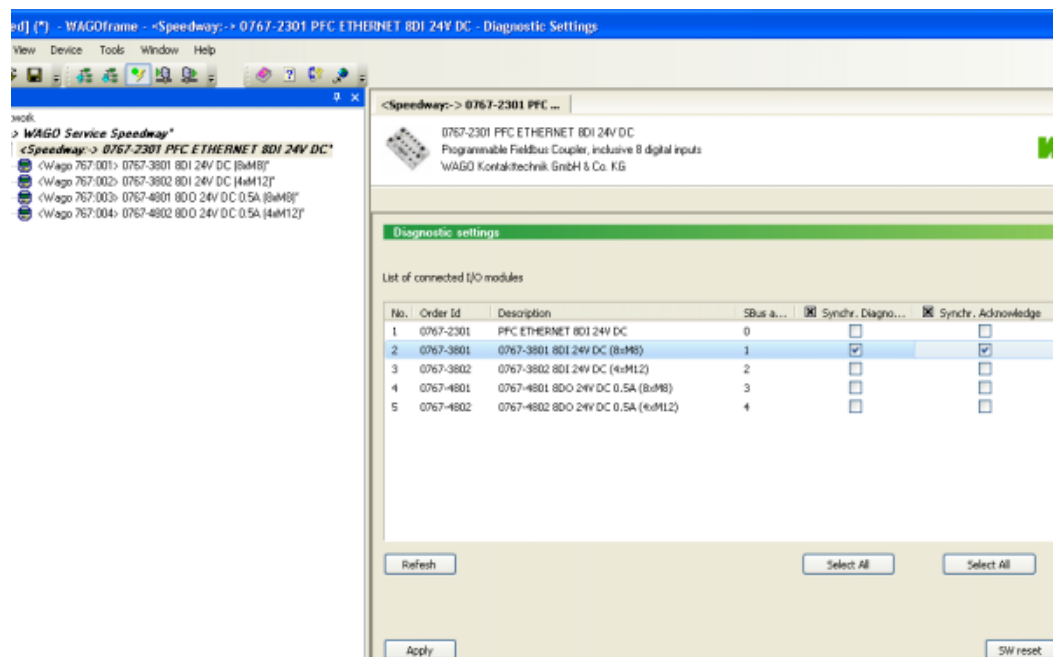


Figure 115: List of connected 767 components

2. Select the synchronous diagnostics and the diagnostic confirmation for the 767 component according to your request by activating or deactivating the selection boxes.

⚠ CAUTION

767 Node Restarting!

Restarting a 767 node can, depending on the parameterization that you performed, place machine components in dangerous circumstances and endanger personnel and machines.

Before pressing **[OK]** to restart, ensure that this will not pose a danger to the machine components.

3. Click **[Apply]** to apply your diagnostic settings. The 767 node is restarted automatically.
4. The diagnostic settings are active when the following text appears in the status indicator: "Application of selected diagnostic settings complete".

16.6.4 Service Page

The Service page is used to restore the default state for selected 767 components.

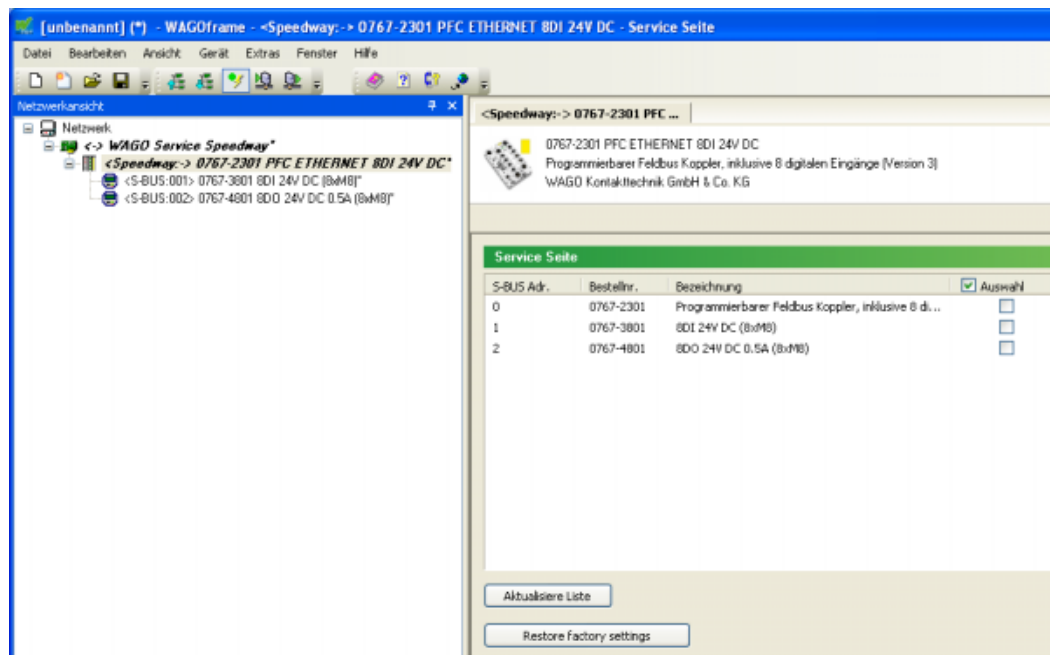


Figure 116: Service page

Table 124: Service page

Parameter/button	Description
S-BUS Addr.	Displays the physical positioni of the devices in the 767 node. 0: Fieldbus coupler 1: First I/O module connected to the fieldbus coupler 2: Second I/O module connected to the fieldbus coupler ...
Item No.	Item number of a 767 component
Designation	Designation of a 767 component
Selection	Selection of the 767 components that should be reset to the default state.
[Update list]	Update the 767 components displayed in the DTM after changing the physical topology (e.g., I/O module removed or added).
[Restore factory]	Reset the selected 767 components to the default state.

16.6.5 User Administration

Use the user administration to change the preset passwords (see Section 11.1) for the users **guest**, **user** and **admin** or set the changed passwords back to the default.

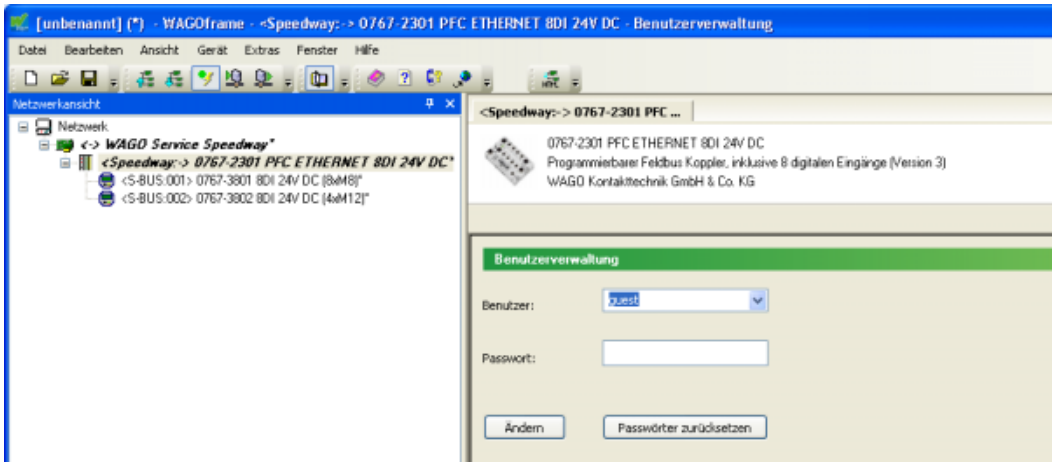


Figure 117: User administration

Table 125: User administration

Parameter/button	Description
User	Select a user whose password should be changed.
Password	Enter a password for the selected user. Only ASCII characters are permitted.
[Change]	Save the new password for the selected user. A dialog appears. Enter admin to confirm the new password and click [OK].
[Reset password]	Reset the password to the default state. This must be performed by a superuser with a special password. Obtain the password indicating the ID listed in the dialog for WAGO Support.



Figure 118: Prompt to reset passwords

16.6.6 File System

Use the file system function to administer the file system on the fieldbus coupler. You can format drives, extract data from the firmware, create or delete directories, upload, download and delete files.

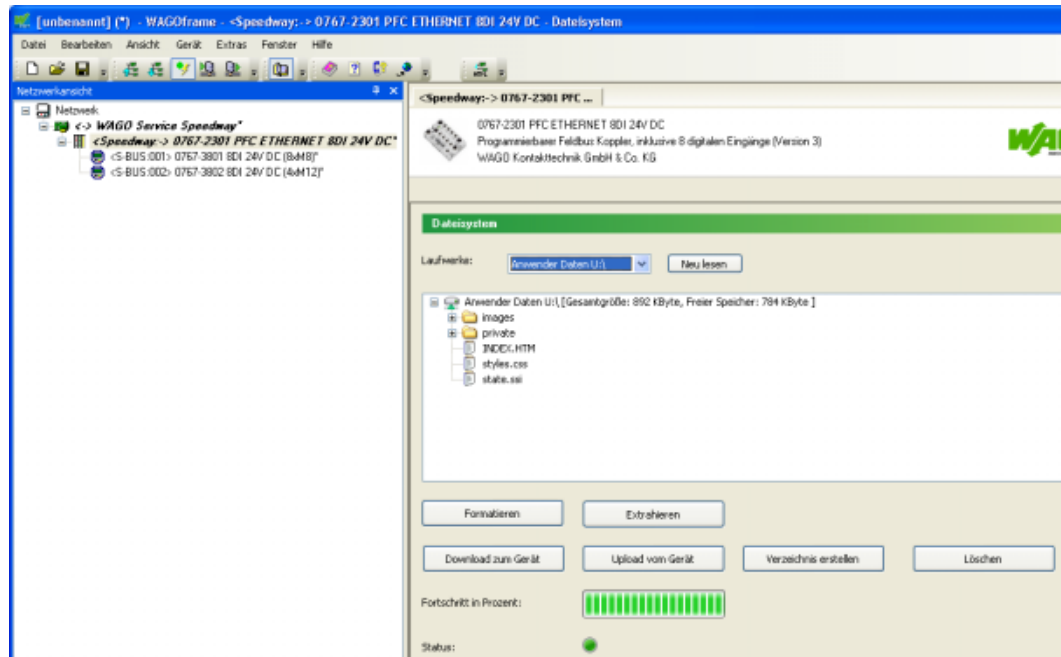


Figure 119: File system

Table 126: File system

Parameter/button	Description
Drives	Select the fieldbus coupler drive: U:\ (WBM pages) P:\ (CoDeSys programm)
[Reread]	Reread the drive currently selected.
[Format]	Format the drive selected under "Drives". All files are deleted.
[Extract]	Extract files from the firmware (WBM, system settings, etc.). Note: Any changes made to the existing files may be overwritten when files are extracted again!
[Download to device]	Copy files from the file system of the PC to the fieldbus coupler. Without selection: Data is saved to the top level of the file system. Selected folder: Data is saved to the selected folder.
[Upload from device]	Copy files from the fieldbus coupler to the PC (e.g., to edit them).
[Create directory]	Create a new director (folder) Without selection: The new directory is created at the top level of the file system. Selected folder: The new directory is created in the selected folder of the file system.
[Delete]	Delete selected files or directories (folders). You can only delete empty directories (folders).
State	Status display of an operation: Green: The selected procedure is completed. Yellow: The selected procedure is active. Red: Error in the action just executed.

16.6.7 Creating a Network

This involves inserting all 767 components into the topology that are connected to the USB connection.

1. Right-click on a coupler in the "Network View" pane or directly on the "WAGO Service Speedway" DTM.
2. In the context menu, select **Scan > Create Network**.
If the "WAGO Service Speedway" DTM is selected, all components connected to the 767 nodes will be carried over into the topology. If your coupler is selected, only the I/O modules will be attached to the coupler.

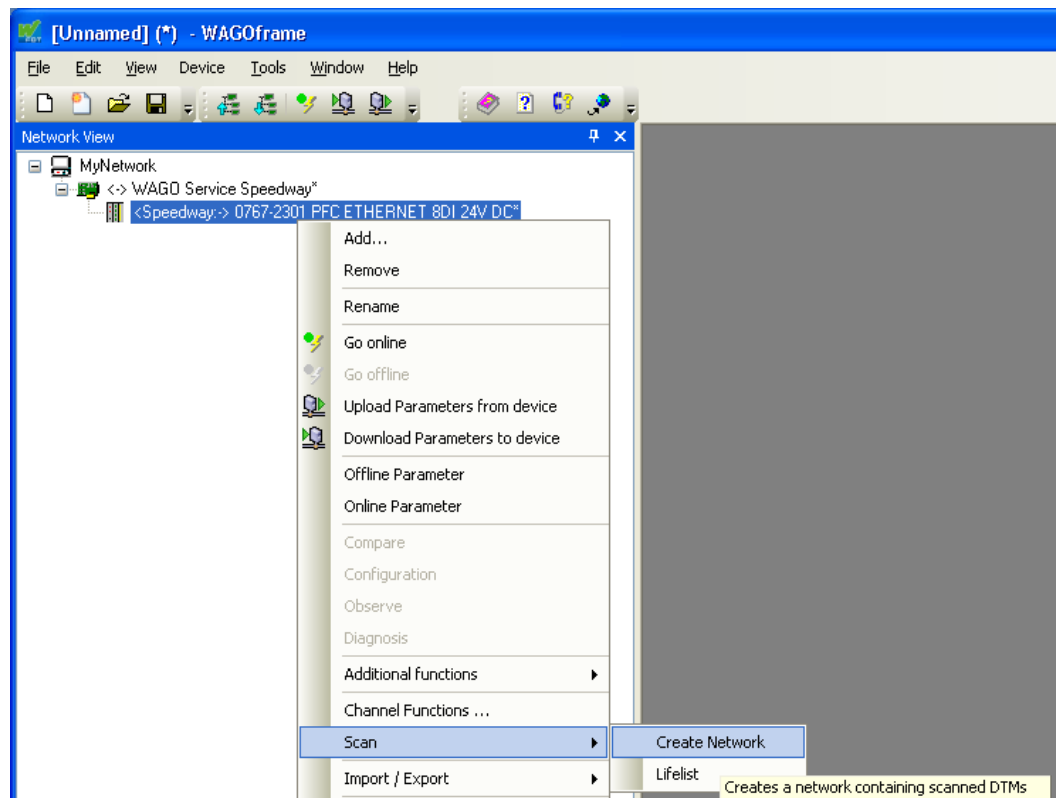


Figure 120: Setting up the network (on I/O modules connected to the fieldbus coupler)

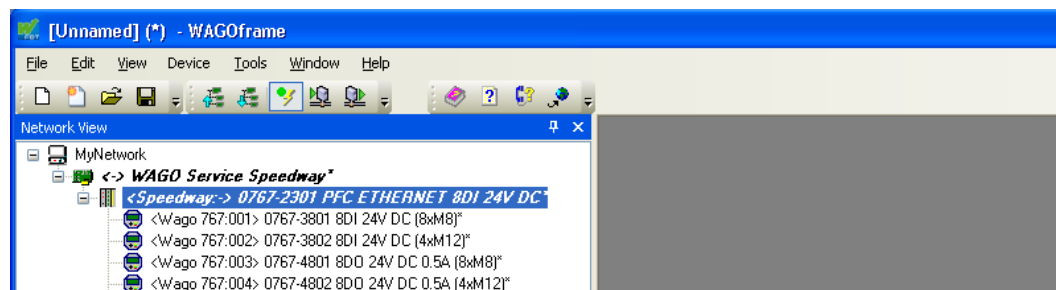


Figure 121: Components connected to the 767 node

16.6.8 Life List

The life list is used to search for all devices connected to the S-BUS in order to select them for configuration directly.

You have the option to use the "WAGO Service Speedway" device driver to search for the connected fieldbus coupler or to use the device driver for the fieldbus coupler to search for I/O modules connected to it.

To search for fieldbus couplers or I/O modules, proceed as follows:

1. Right-click on the respective device driver in the "Network View" pane. In this example, the device driver for the fieldbus coupler is selected to search for the I/O modules connected to it.
2. In the context menu, select **Scan > Life List**. The life list opens.

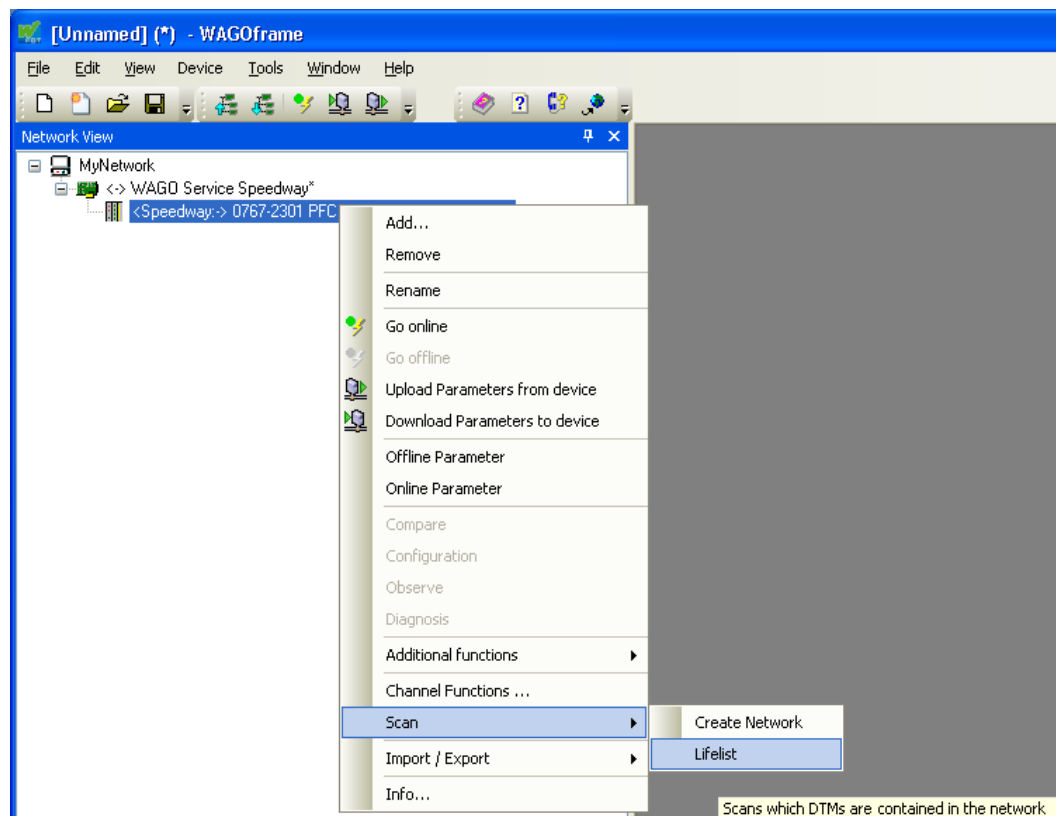


Figure 122: Searching for connected I/O modules using the life list

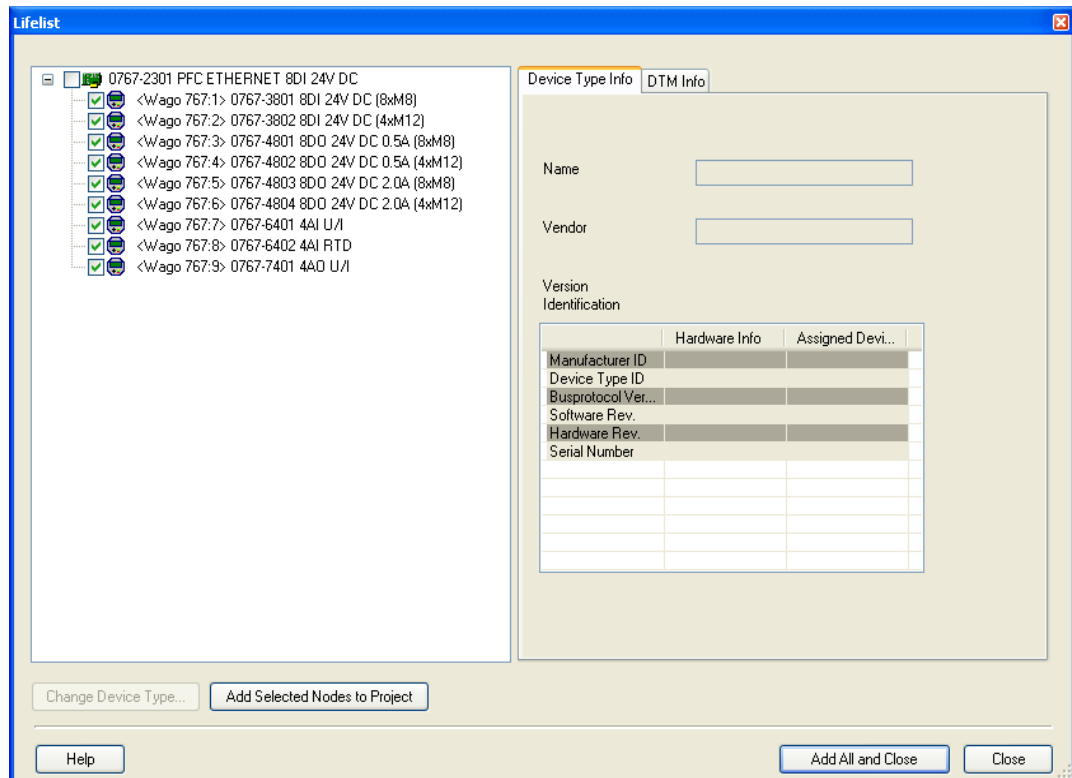


Figure 123: Connected I/O modules of the selected fieldbus coupler

- Now select from the list those devices that you wish to carry over into the "Network View" for configuration. Click **[Add Selected Nodes to Project]** and then **[Close]**.

16.6.9 System Update

Firmware update for the 767 Series components is performed via system update. To ensure that the fieldbus node remains consistent and executable after updating the firmware, system update must be performed for both fieldbus coupler and connected I/O modules.

NOTICE

System update

Before updating, observe the following measures to prevent any possible damage to the 767 system:

- The power supply must not be disconnected while updating!
 - To exclude any interferences by the fieldbus, the fieldbus cable must be disconnected before updating!
-

Requirement:

- You have installed the DTM (759-370) WAGOframe
- You have installed the DTM (759-371) WAGO service interface
- You have installed the DTM (759-922) USB driver
- You have installed the DTM (759-362) system update.
- Update packages are available for the connected 767 Series components.

System Update Procedure

To perform a system update for each 767 Series component, please complete the following steps:

1. Read 767 components' parameters and save them on your PC.
2. Update 767 Series components' firmware.
3. Rewrite parameters from your PC to the 767 Series components.
4. Set parameters to valid and finish the procedure.

16.6.9.1 Adding the DTM System Update

To add the DTM system update to WAGOframe, please proceed as follows:

1. In "Network View", right-click on the "WAGO Service Speedway" device driver.
2. In the **Add...** context menu, select "Add". The "Add" dialog box opens.

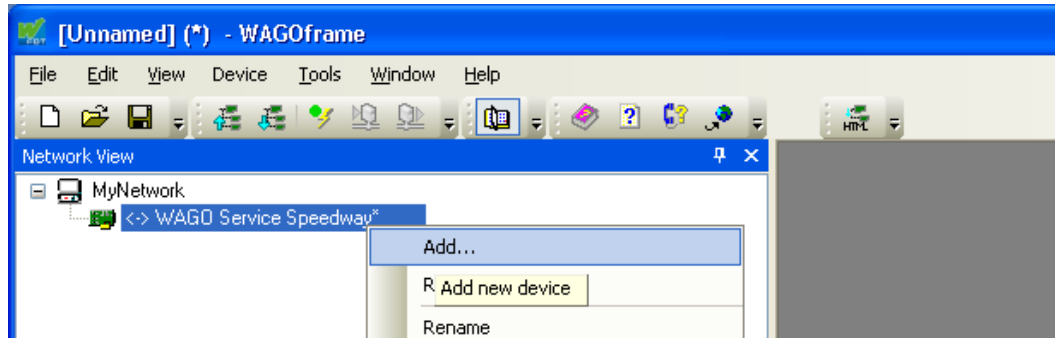


Figure 124: Adding the DTM system update

3. In the "Add" dialog box, select the DTM **0767 System Update**.
4. Click **[OK]** to confirm your selection.

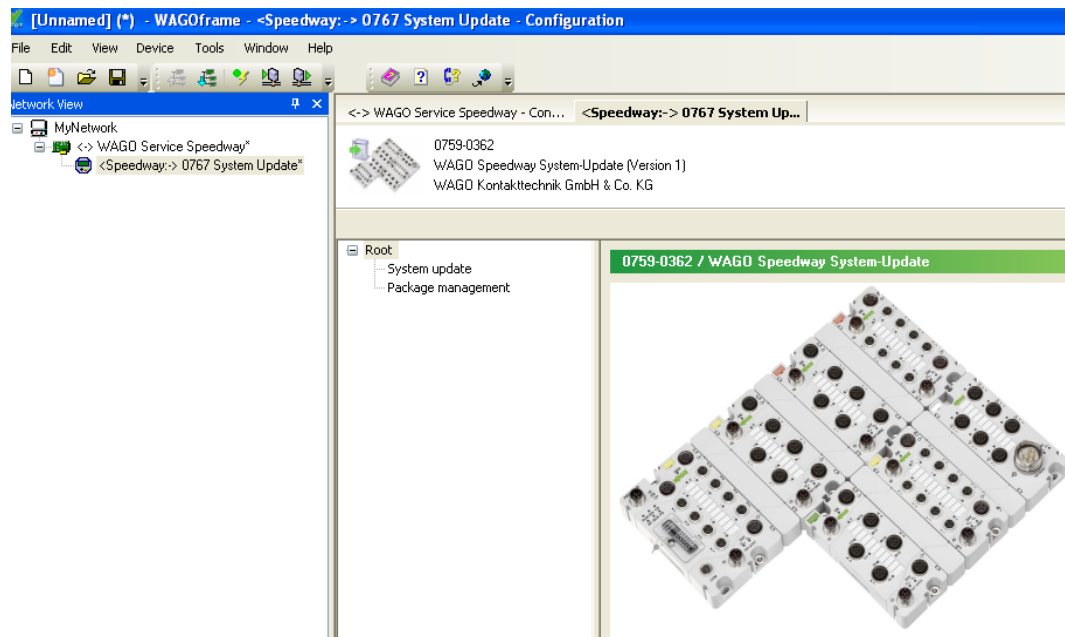


Figure 125: DTM system update

16.6.9.2 Go online to 767 Nodes using Update DTM

The firmware can only be updated when a communication connection exists between Update DTM and 767 node. Please proceed as follows:

1. In the "Network View", right-click on the "<Speedway:> 0767 System Update" device driver.
2. In the context menu, select **Go online**. When the progress bar displays 100% and the entry is displayed in ***bold italics***, the communication connection is established.

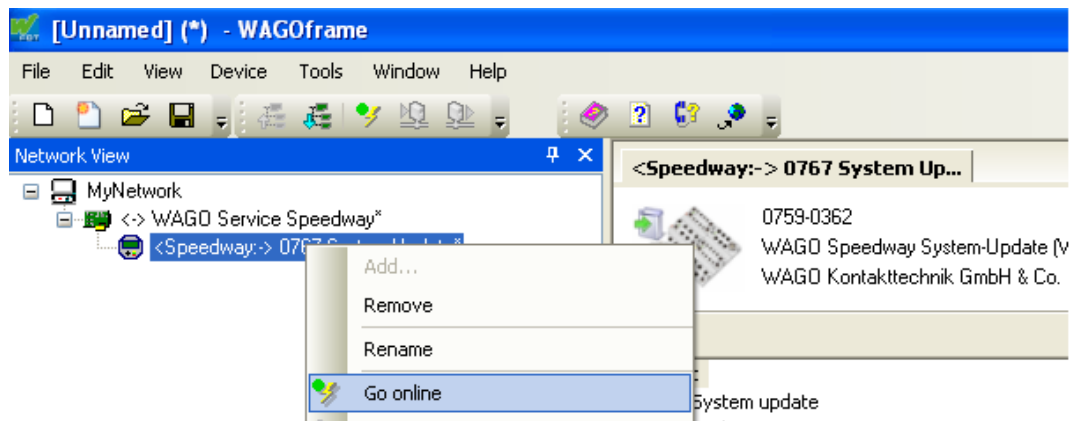


Figure 126: Go online to 767 node

16.6.9.3 Updating the 767 Components

The current firmware is available from WAGO Support. Send an e-mail with the subject "Current Speedway Firmware" and the item number of the respective 767 components to: support@wago.com.

Import firmware packages

To use the received firmware packages, import them into the DTM system update. Please proceed as follows:

1. Save the received firmware packages with the "*.wup" extension to any directory on your PC.
2. Open the DTM user interface by double clicking "**0767 System Update**" in the network view.

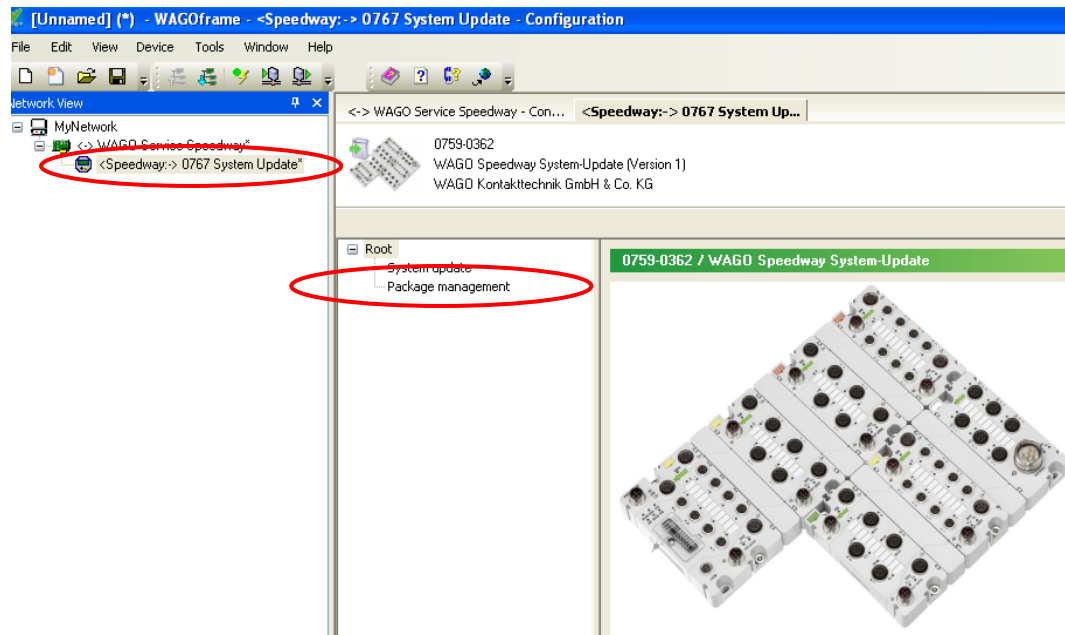


Figure 127: Package management 1

3. Click "Package Management" in the left window of WAGOframe.
4. To import the received firmware files, click **[Import]**. In the window that opens, go to the directory where you have saved the firmware files and select the file to be used. Click **[Open]** to apply the files.

Delete firmware packages

To maintain a clear "Package Management" interface, you can remove unneeded update packages from the view. Please proceed as follows:

1. Select the required firmware files in the right window.
2. Click **[Delete]** to remove the selected firmware packages.

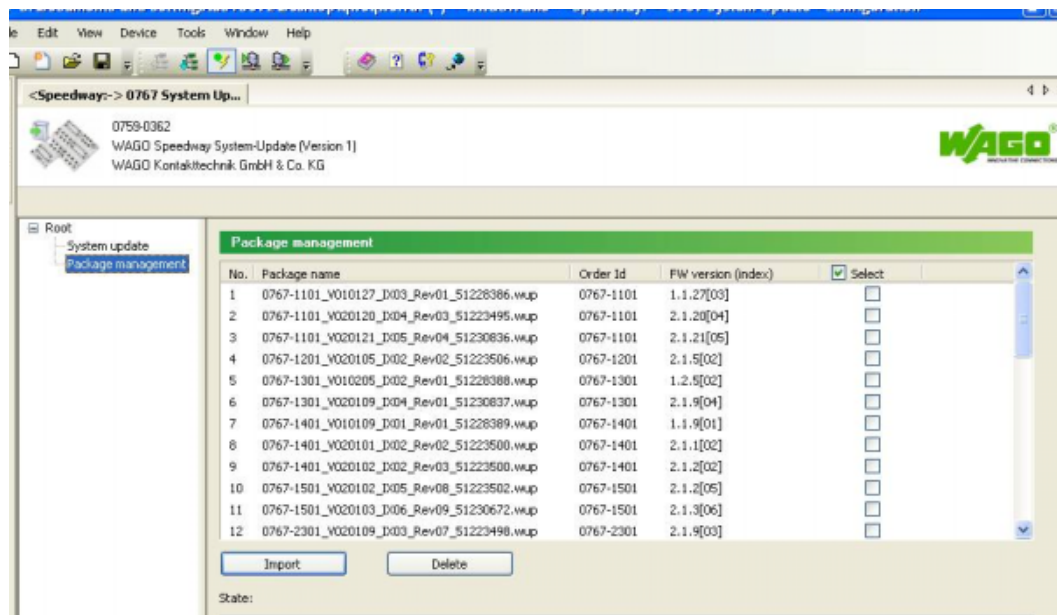


Figure 128: Package management 2

System update



Note

When updating the firmware of the fieldbus coupler, the saved module parameters may be overwritten. Therefore, check your existing configuration after updating the firmware.

Perform the system update here. The module settings you made normally remain unchanged. Otherwise, a corresponding warning message appears. If you still want to update the firmware, the 767 components are returned to their default state.

1. Click "System Update" in the left window.

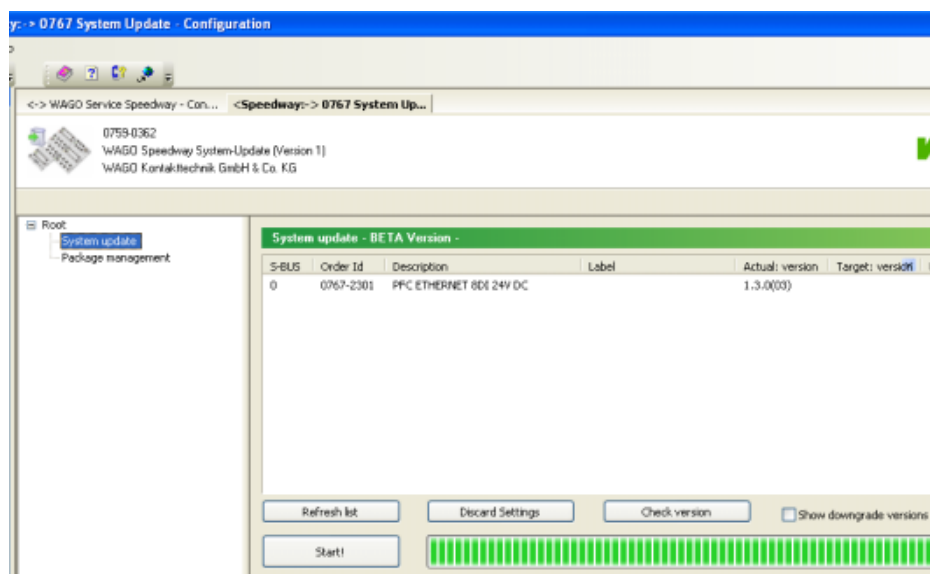


Figure 129: System update 1

2. The fieldbus coupler with all connected I/O modules are listed in the right window. All 767 components that can be updated are pre-selected. If the pre-selection is incorrect or if specific 767 components should not be updated, deselect them ("Update?" column).
 "Actual: Version": Firmware currently present on the device
 "Target: Version": Version of the firmware that should be loaded into the 767 components. If multiple "Target" versions can be selected, select the one relevant to you.

- Click **[Start!]** to update the system. The 767 components are marked in yellow while being updated.

Note



During the firmware updating process, the fieldbus coupler disconnects each of its COM ports. A PC equipped with Windows 2000 will detect this, and a Windows message appears. This is not an error message. Confirm the message by pressing **[OK]**.

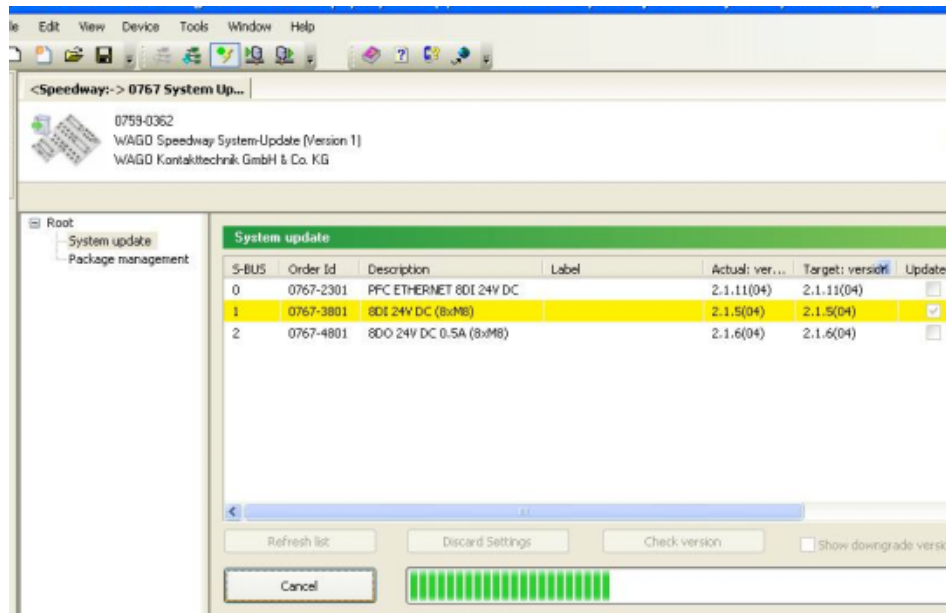


Figure 130: System update 2

Table 127: Buttons

Button	Description
[Update List]	Use this function to read the node design and the view updates.
[Discard Settings]	Discard the selections and settings performed by you.
Display downgrade versions	If this checkbox is selected, the versions to downgrade a device are also displayed in the list of target versions.
[Start!]/[Abort]	Start/abort system update.

4. If the system update is finished, the updated 767 components are displayed in green (see figure).

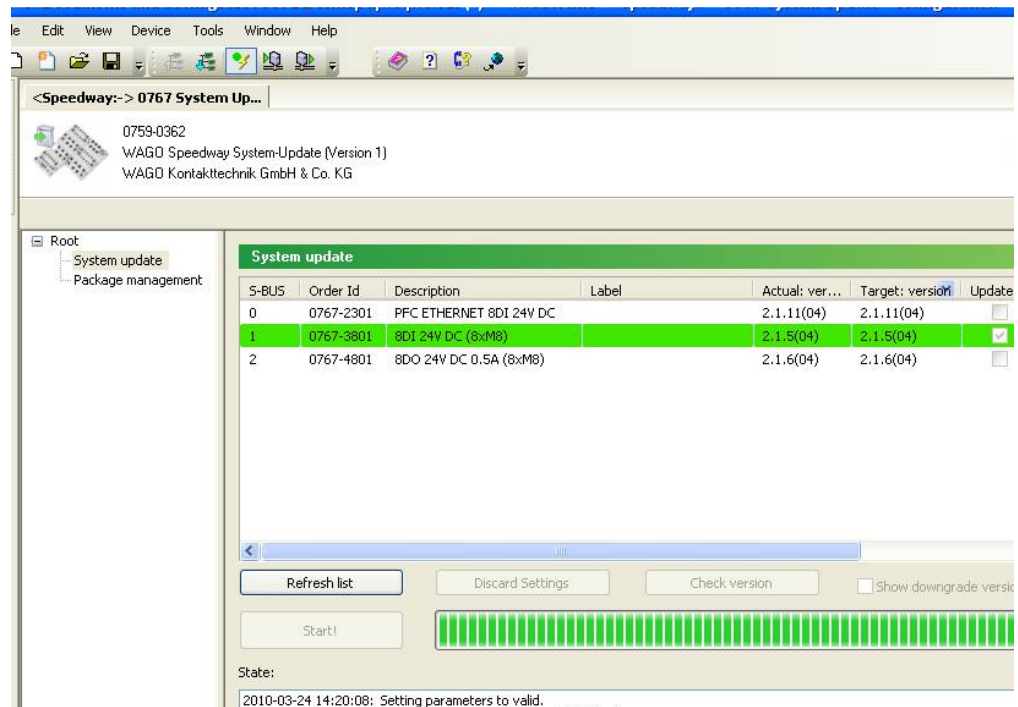


Figure 131: System update 3

During system update, all relevant information is stored on your PC. Update can be repeated subsequently if the system update fails (components are displayed in red). In this case, original parameters remain unchanged.

If the update for a device fails, please contact WAGO Support.

16.7 Parameterization

All parameters listed here can be set using WAGOframe (or another FDT/DTM border application) for the fieldbus coupler. If you use a fieldbus for the parameterization, only certain parameters are configurable, depending on fieldbus type.

The following Sections contain information on the parameters and their descriptions.

To open the parameterization user interface (DTM) of a 767 component, double-click on the appropriate 767 component in the "Network View". The parameterization user interface can also be opened by right-clicking on **Offline Parameter** or **Online Parameter** in the context menu.

If several parameterization user interfaces are open, select one via the corresponding tabs. If you switch a 767 component from the online mode to the offline mode or vice versa, close the parameterization user interface and reopen it.

Depending on the selection of parameterization user interfaces, various buttons are at your disposal:

Table 128: DTM buttons

Button	Description
[Read] (Online mode only)	Reads and displays the parameters found in the 767 components.
[Write] (Online mode only)	Writes the modified values to the 767 components.
[Close] (Online and offline mode)	Closes the parameterization user interface (DTM).
[Apply] (Offline mode only)	Applies the entries in the project. Please note that the project should also be subsequently saved (File > Save).
[Help] (Online and offline mode)	Opens the online help for an entry that has been previously selected in the DTM (e.g., digital inputs, global setting).
	Shows/hides parameter overview.
	Displays the product data sheet. A PDF reader must be installed on your PC.
	Opens the DTM online help.

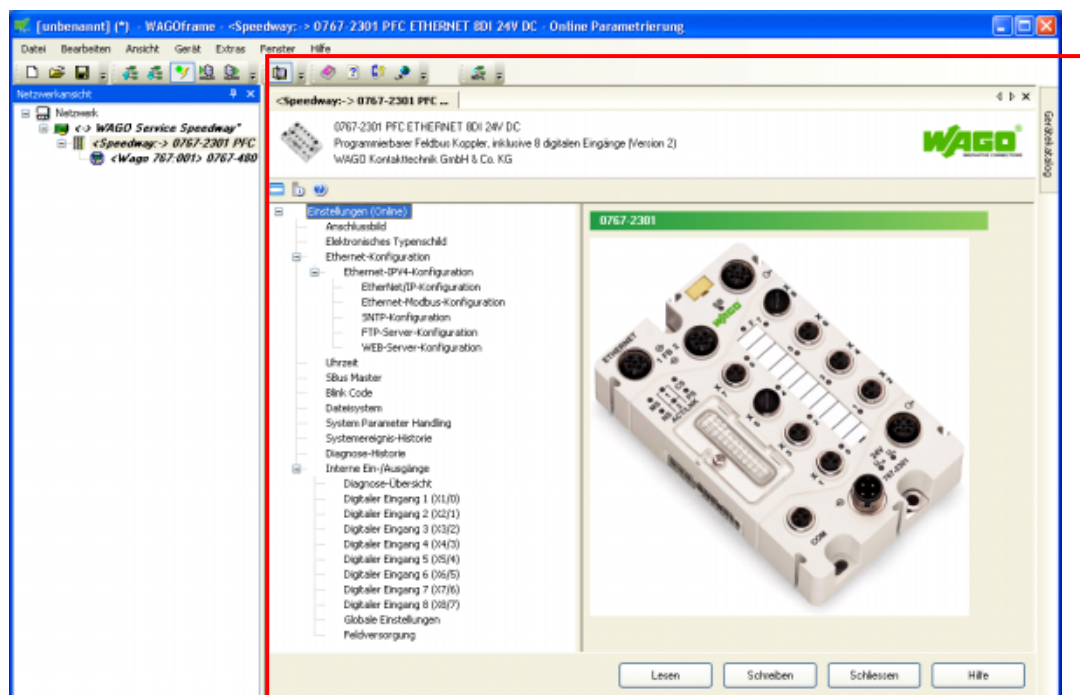


Figure 132: Example of an opened DTM with the available parameters

16.7.1 General Parameters

Electronic type label

Table 129: Information about the fieldbus coupler

Parameter	Description
Vendor	Manufacturer
Release index	SW.HW.FL FW: Actual firmware release index When updating the firmware, please note that the firmware release index may not be conformed to the printed firmware release index on the side of the fieldbus coupler. HW: Hardware release index FL: Firmware loader release index
Firmware revision	General information on the module
Order number	
Description	
Serial number	
Date of production	
Designation	Electronic marking field (max. 40 characters)

Date/Time

Table 130: Time settings

Parameter	Description
Daylight saving time	Activate or deactivate summer time here.
Time zone	Select your time zone here.
Actual time	Set the date and time here.
UTC	World time display

To transfer the current local time from your PC to the fieldbus coupler, click **[Apply PC time]** and then **[Write]**.

DIP switch

Display of the switch positions of the existing DIP switches on the fieldbus coupler.

S-BUS Master

Table 131: S-BUS-Master

Parameter	Description
Max. restarts	In the input field, enter how often the fieldbus coupler should start up again following a S-BUS disruption. 0x0: No restart 0x1 – 0xFFFFFFFF: Number of restarts 0xFFFFFFFF: Endless restarts
Actual baudrate	Actual baudrate display
Actual S-Bus cycle time	Actual S-Bus cycle time display

Blink Code

Table 132: Display of errors

Parameter	Description
Blink code	Display of blink code (error group, error code and error argument).

A list of blink codes and error descriptions, as well as information on error correction, can be found in Section 15.3.3.

System Parameter Handling (SPH)

The SPH identifies changes to the configuration of a 767 node and to the automatic configuration of the I/O modules. When a configured I/O module must be replaced due to a defect, you do not need to reconfigure the new I/O module. The stored parameters are automatically transferred to the new I/O module.

Requirement: The exchanged I/O module must have the same item number as the previous one.

To activate the SPH function, proceed as follows:

1. Select "Create nominal system configuration" to store the currently set parameters of the I/O modules in the file system of the fieldbus coupler (parameter database is created).
2. If you consider the firmware status of the new I/O module relevant, select the "FW/HW Release index check I/O modules" checkbox. This requires that the exchanged I/O module has the same firmware and hardware status as the defective I/O module.
If the firmware status is deviant, a blink code is emitted in the "Error Code" field and on the CS LED of the fieldbus coupler. In this case, the firmware should be updated. For more information, see Section 16.6.9.

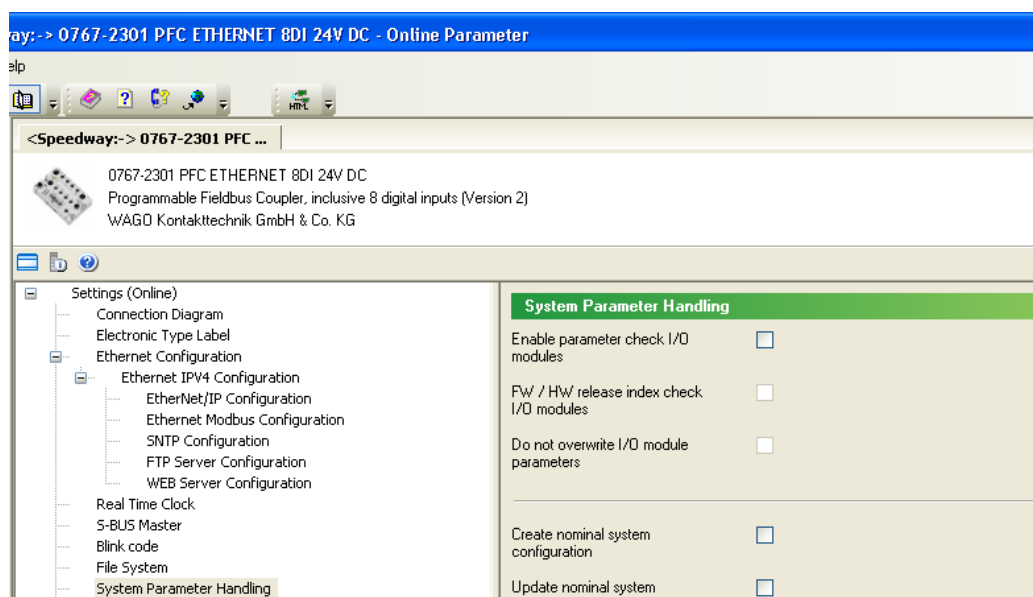


Figure 133: System parameter handling 1

3. Select "Enable parameter check I/O modules" to enable automatic configuration. The dialog box below appears.
Click **[YES]** if you have not created a database yet.
Click **[NO]** if you have already created a database (step 1). Otherwise, you will overwrite the existing database.

The stored parameters are now compared with the current parameters at each start-up. If this process reveals that parameters differ, these parameters are automatically overwritten by the stored values in the I/O module.

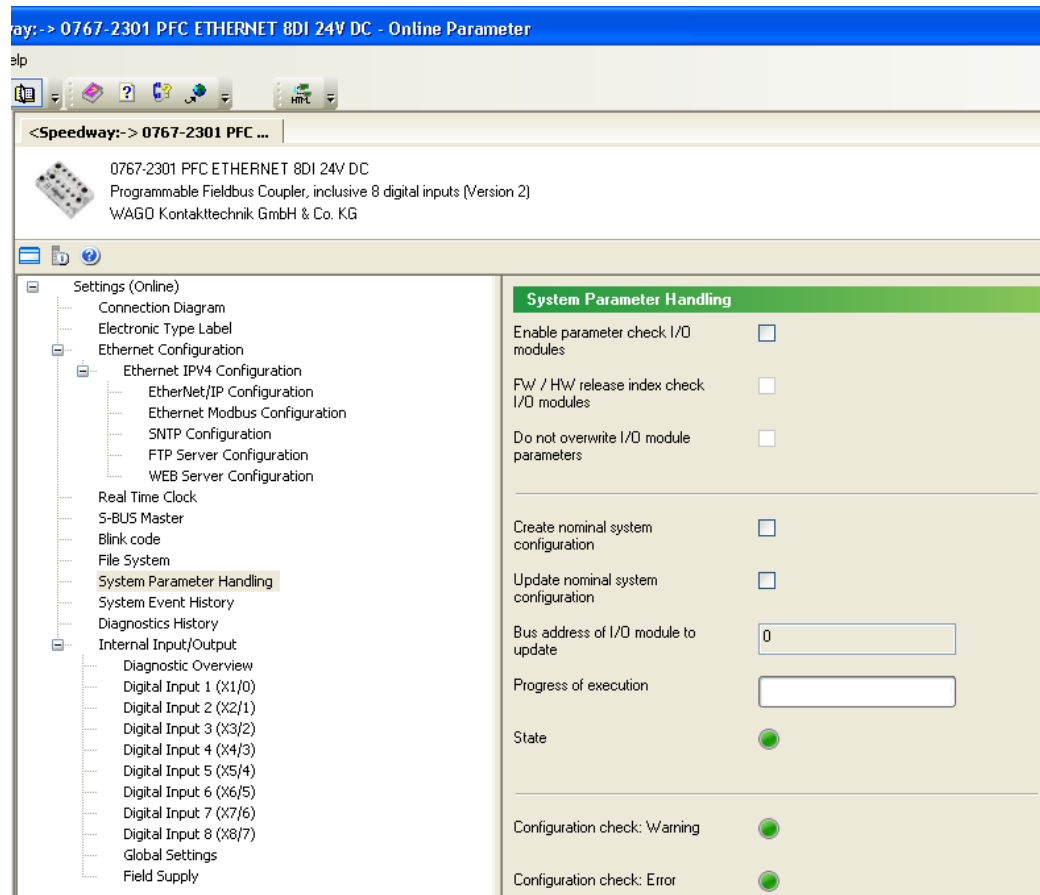


Figure 134: System parameter handling 2

Table 133: Overview of adjustable parameters for "System Parameter Handling"

Parameter	Description
Enable parameter check I/O modules	Select this checkbox to enable the parameter check for the I/O modules and for the integrated inputs of the fieldbus coupler.
FW/HW release index check I/O modules	Select this checkbox to check the FW/HW release index of the IO modules. The checkbox is only activated when "Enable parameter check I/O modules" is checked.
Do not overwrite I/O module parameters	Select this checkbox to prevent an automatic overwrite of the I/O module parameters. The checkbox is only activated when "Enable parameter check I/O modules" is checked.
Create nominal system configuration	Select this checkbox to create the parameter database.
Update nominal system configuration	Select this checkbox to update the system configuration for an I/O module. To do this, select the address of the I/O module from the "Bus address of I/O module to update" field. To update the I/O area of the coupler, select address 0.
Bus address of I/O module to update	Enter here the bus address of the I/O module that is to be updated. The checkbox is only activated when "Enable parameter check I/O modules" is checked. The first I/O module on the fieldbus coupler has bus address 1, the second has bus address 2, etc.
Progress of execution	The progress of the "Create nominal system configuration" and "Update nominal system configuration" parameters is shown here.

Table 133: Overview of adjustable parameters for "System Parameter Handling"

Parameter	Description
State	The status of the "Create nominal system configuration" and "Update nominal system configuration" parameters is shown here: Green: Procedure OK Yellow: Procedure active Red: Error
Configuration check: Warning	The warning message is displayed when the "I/O module parameter not overwritten" parameter is activated, but a change between stored and current parameters has been detected. Green: No warning during configuration check Red: Warning during configuration check
Configuration check: Error	An error message is shown here when an error is discovered during the configuration check or the automatic configuration (see Section 15.3.3): Green: No error during configuration check Red: Error during configuration check
Error code	The error code is displayed here for the configuration check error or warning that has occurred. The display is only activated when a warning or error has arisen.
Bus address of first failed component	The bus address of the first failed I/O module is displayed here. The display is only activated when a warning or error has arisen. If the error cannot be localized, the address "65535" (hex.: 0xFFFF) is displayed.

System Event History

The system event history contains a display of all system events from the fieldbus coupler. These events are also displayed as a blink code via the CS LED (see Section 15.3.3). The system events are permanently saved in the fieldbus coupler.

Table 134: Overview of system event history

Parameter	Description
Memory overflow	The status of the circular buffer is displayed here. Green: Circular buffer OK Red: Circular buffer overflow (old events are overwritten)
Entries	All events are displayed in the table. No.: Ongoing number of events Error: Notification type (Incoming: incoming notification Outgoing: outgoing notification) Error code: Error code (see Section 15.3.3) Data: Additional information on the error code Counter: Event counter (increases with repeated occurrence of identical events) Time stamp: Occurrence of events in local time of fieldbus coupler
Clear entries	Select this checkbox to delete all events.

Diagnostics History

All diagnostics of the fieldbus coupler and I/O modules are stored in the diagnostics history. The collective diagnostics are not permanently saved.

Table 135: Overview of diagnostics history

Parameter	Description
Entries	The entries of the diagnostics history are displayed in the table. The last chronological entry is always first in the table. ID: Current number of diagnostic entry S-BUS address: Bus address of the fieldbus coupler/I/O module Param. add.: Parameter address Time stamp: Occurrence of diagnostics in local time of fieldbus coupler Data: Additional information on the diagnostics
Clear entries	Input here the number of entries to be deleted. The oldest entries are deleted first. When the quantity entered is greater than the current number of entries, all entries are deleted.
Stop update	Stop the cyclic reading of the diagnostic messages.
ID of last entry	The consecutive number of the last entry in the history is displayed here.
Maximum size of history	The maximum size of the diagnostics history is displayed here in bytes.
Current history size	The current size of the diagnostics history is displayed here in bytes.

16.7.2 Fieldbus-Specific Parameters

Ethernet Configuration

Table 136: Parameters for Ethernet configuration

Parameter	Description
MAC ID	Information that is only displayed when the fieldbus coupler is in online mode.
Interface	
Host name (active)	
Domain name (active)	
Host name (device storage)	Enter here the host name that is to be used when restarting the fieldbus coupler.
Domain name (device storage)	Enter here the domain name that is to be used when restarting the fieldbus coupler.

Ethernet IPV4 Configuration

Table 137: Parameters for configuring the Ethernet IPV4

Parameter	Description
IP address (active)	Information that is only displayed when the fieldbus coupler is in online mode.
Subnet mask (active)	
Gateway address (active)	
IP address DNS 1 (active)	
IP address DNS 2 (active)	
IP address from ...	Select here the assignment of the IP address: - Device memory - BootP - DHCP Note: If you use the DIP switch to set an IP address, the options listed here for assigning the IP address are irrelevant!
IP address (device storage)	Enter here the static IP address that is to be used when restarting the fieldbus coupler. "Device memory" must be selected for this to take place (see parameter "IP address obtained from ...").
Subnet mask (device storage)	Enter here the static subnet mask that is to be used when restarting the fieldbus coupler.
Gateway address (device storage)	Enter here the gateway address that is to be used when restarting the fieldbus coupler.
DIP switch IP address	Switch on the first 3 bytes of the IP address here when you use the DIP switch for assigning an IP address.
IP address DNS1 (device storage)	Enter here the DNS server address that is to be used when restarting the fieldbus coupler.
IP address DNS2 (device storage)	Enter here the address of an alternative DNS server that is to be used when restarting the fieldbus coupler.

Ethernet IP Configuration

Table 138: Parameters for configuring the Ethernet IP

Parameter	Description
Enable EtherNet/IP server	Activate and deactivate the EtherNet/IP protocol here.
Field bus error reaction	Set the desired response of the fieldbus coupler to a fieldbus error here. Selection box: - Set outputs to 0 Sets the outputs of the I/O modules to 0 - Keep last values Retains the last values of the outputs of the I/O modules
S-BUS error reaction	The response of the fieldbus coupler to an S-BUS error appears: - Set input values to 0 (sets the process value of the input modules to 0)
Bytes of fieldbus inputs added to Assembly Inst. 101-103, 113	Here you can set the number of bytes for the fieldbus input variables that are to be added to assembly instances 101 through 103 and 113. Value range: 0 – 512
Bytes of fieldbus outputs added to Assembly Inst. 104-109, 112	Here you can set the number of bytes for the fieldbus output variables that are to be added to assembly instances 104 through 109 and 112. Value range: 0 – 512
Bytes of fieldbus inputs received via Assembly Inst. 111	Here you can set the number of bytes for the fieldbus input variables that are to be received via assembly instance 111. Value range: 0 – 512
Byte Offset of fieldbus inputs received via Assembly Inst. 111	Here you can set the number of bytes for the fieldbus output variables that are to be received via assembly instance 111. Value range: 0 – 512
Bytes of fieldbus outputs sent via Assembly Inst. 110	Here you can set the start position of the fieldbus input variables that are to be sent via assembly instance 110. Value range: 0 – 512
Byte Offset of fieldbus outputs sent via Assembly Inst. 110	Here you can set the start position of the fieldbus output variables that are to be sent via assembly instance 110. Value range: 0 – 512
RUN/IDLE header originator to target	With this parameter, you can activate the RUN/IDLE header from the sender to the receiver.
RUN/IDLE header target to originator	With this parameter, you can activate the RUN/IDLE header from the sender to the receiver.
ODVA vendor id	40 (WAGO)
ODVA product code	2301
ODVA serial number	Unique serial number for product identification
ODVA product name	WAGO PFC ETHERNET 8DI 24 V DC

Ethernet MODBUS Configuration

Table 139: Parameters for configuring the Ethernet MODBUS

Parameter	Description
Modbus version (device memory)	MODBUS version change (compatibility mode for Release 1) - Variant 1 (Release 1) - Variant 2 (Release 3)
Modbus version (active)	Display of which MODBUS version is active.
Enable TCP server	Disable or enable the MODBUS/TCP server.
Enable UDP server	Disable or enable the MODBUS/UDP server.
Connection watchdog	Enter here the time duration after which the MODBUS/TCP connection is to be closed if communication is not established during this period. <i>Default setting: 60000 ms</i>
Close connection after timeout	You have the following setting options: 0: - 1: After the watchdog has timed out, the MODBUS/TCP connection is automatically closed. <i>Default setting: 0</i>
Watchdog type	Select the watchdog type. 0: With function code trigger 1: Function code trigger not relevant <i>Default setting: 0</i>
Watchdog time	Set the time for monitoring (100 ms step). <i>Default setting: 100 ms</i>
Function code triggers 1 – 16	Reset the watchdog using the MODBUS function codes 1 – 16. <i>Default setting: 65535 (0xFFFF). All function codes are relevant.</i>
Function code triggers 17 – 32	Reset the watchdog using the MODBUS function codes 17 – 32. <i>Default setting: 65535 (0xFFFF). All function codes are relevant.</i>
Watchdog status	The current watchdog status is indicated here. 0: Not active 1: Active 2: Elapsed

SNTP Configuration

Table 140: Parameters for configuring the SNTP

Parameter	Description
Enable SNTP	Wählen Sie hier aus, ob Sie die Systemzeit über SNTP automatisch aktualisieren möchten. Select here whether you would like the system time to be automatically updated via the SNTP.
Update interval	Set the length (in s) of the synchronization interval here.
Server timeout	Select here the maximum time (in ms) that may elapse before the server responds to an SNTP request.
IP address SNTP server (active)	Displays the IP address currently used for the SNTP server.
IP address SNTP server (device storage)	Enter an IP address for the SNTP server, that is to be used when restarting the fieldbus coupler.

FTP Server Configuration

Table 141: Parameters for configuring the FTP server

Parameter	Description
Enable FTP server	Activate or deactivate the FTP server here.

WEB Server Configuration

Table 142: Parameters for configuring the WEB server

Parameter	Description
Enable WEB server	Activate or deactivate the WEB server here.
Port number	Communications port for the Web server. Default setting: 80

16.7.3 Parameters of Inputs/Diagnostic Overview

Diagnostic Overview

The currently pending diagnostics existing on the module are displayed here. In this view of the DTM, you can enable simulation of the diagnostics, as well as disable transmission of the diagnostics. When disabling transmission, make sure that the display behavior of each LED changes that indicates the specific diagnostics (Section 15.2). The diagnostic overview is only available in online mode.

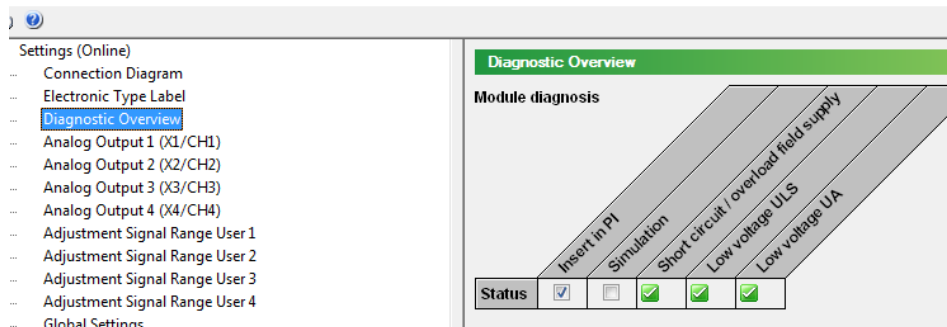


Figure 135: Example of the diagnostic overview of a module (information may differ from the actual module)

Table 143: Diagnostics setup

Parameter	Description
Insert in PI	Displays whether there is a diagnostics: X mark: There is a diagnostic message. Check mark: There is no diagnostic message.
Simulation	Displays whether there is a diagnostics: X mark: There is a diagnostic message. Check mark: There is no diagnostic message.
Status	Displays whether there is a diagnostics: X mark: There is a diagnostic message. Check mark: There is no diagnostic message.

Table 144: Information about existing module diagnostics

Modul Diagnostics	
Diagnostics	Description
Short circuit/overload field supply	The device has detected a short circuit or overload of the field supply (only possible when field supply is switched on).
Low voltage ULS	If an undervoltage of the logic or sensor supply (U_{LS}) of $< 18\text{ V}$ occurs on the device, a corresponding diagnostic message is generated and the F-LED illuminates. The substitute value strategy that you have parameterized is used on the digital inputs.
Low voltage UA	If an undervoltage of the actuator supply (U_A) of $< 18\text{ V}$ occurs on the device, a corresponding diagnostic message is generated and the F-LED illuminates. The undervoltage of the actuator supply has no functional effect on the device.

Parameters of Inputs

Table 145: Overview of adjustable parameters for the digital inputs

Parameter	Description
Designation	Electronic marking field (max. 40 characters) to give the channel of the fieldbus coupler a meaningful name, e.g. "Light switch in section A".
Input value	The current input signal is displayed here. If the "Simulation Input Value" parameter has been selected, select here the input value that is to be simulated: If simulation is switched off* Checkbox unselected: Input value (0) Checkbox selected: Input value (1) If simulation is switched on Checkbox unselected: Input is simulated with value 0 Checkbox selected: Input is simulated with value 1
Signal inverting	You can invert the currently pending input signal here. Checkbox unselected*: Input signal is reproduced in the process image in the same manner as at the input Checkbox selected: Input signal is reproduced in the process image in an inverted manner
Substitute Value Strategy	This releases the substitute value or the last input value in cases such as a fieldbus interruption. You have the following options: - Switch to substitute value* - Retain last value**
Substitute Value	Enter here the process value that is used in case of an error. In the case of an error (fieldbus interruption), this value is used with the "Switch to Substitute Value" substitute value strategy. Checkbox unselected: 0* Checkbox selected: 1
Filter time	Set the input filter for the measured signals here. You have the following options: - none - 0.1 ms - 0.5 ms - 3 ms* - 15 ms - 20 ms

Table 145: Overview of adjustable parameters for the digital inputs

Parameter	Description
Simulation input value	This simulation can be used to simulate input values. Checkbox unselected*: The value at the inputs is carried over into the process image. Checkbox selected: The simulation value is inserted into the process data independent of the input value. The "Input Value" parameter is selected.

* Default setting

16.7.4 Global Settings

Table 146: Overview of parameters for the entire fieldbus coupler

Parameter	Description
Simulation diagnostic	If the checkbox is selected, you can simulate a low voltage diagnostic. To generate a low voltage diagnostic, one or both of the two checkboxes "Low voltage ULS" and "Low voltage UA" must be selected. <i>Default setting: unselected</i>
Low voltage ULS	In the case of an undervoltage of the logic and sensor supply (ULS) or the actuator supply (UA), the corresponding diagnostic is displayed here.
Low voltage UA	

16.7.5 Parameters of Field Supply

Table 147: Overview of adjustable parameters for the field supply

Parameter	Description
Enable field supply	Switch on the field supply (24VDC) for the sensors here. <i>Default setting: selected</i>
Autorestart delay	In the event of a short circuit, the sensor supply is switched off for a parameterizable amount of time. Here, enter the delay time (in 100-ms increments); after this time, the sensor supply is restarted. If the short circuit still exists, the process is repeated.
Simulation diagnostic	The simulation can be used to simulate a short circuit. <i>Default setting: unselected</i>
Short circuit/overload	If simulation is deactivated, the respective error is displayed upon emergence. If simulation is activated, you can simulate one of the errors by selecting the appropriate parameter.

17 Maintenance and Service

This section contains information on maintenance and service.

17.1 Updating the Firmware

Information on updating the firmware can be found in Section 16.6.9.

17.2 Replacing the Fieldbus Coupler

To replace the fieldbus coupler, e.g., to change variants, proceed as described in the following sections.

17.2.1 Disconnecting the Cables

Before removing the connectors, clean the fieldbus coupler to ensure that no dirt or other material comes in contact with the connections. This can lead to damage of the contacts.

To unplug the cables, proceed as follows:

1. Disconnect the power supply from those devices on which you have mounted the fieldbus coupler.

CAUTION

Hot connection sockets!

Even when taking into account derating, high surface temperatures on the metallic connection sockets and on the enclosure can arise during operation. If the 767 Series component has been in operation, allow it to cool before touching or use gloves.

2. Unscrew all screw connections and remove the cables.

17.2.2 Removing the fieldbus coupler from your system

To remove the fieldbus coupler from the framework of your system, proceed as follows:

1. Disconnect the power supply from those devices on which you have mounted the fieldbus coupler.
2. Release the fieldbus coupler from the framework of your system by unscrewing the M4 screws.

17.2.3 Removing the Fieldbus Coupler from the Carrier Rail

In order to keep a clear representation, the mounting rail adapter in the figure below (B, C) is shown without the fieldbus coupler.

If the fieldbus coupler is mounted on a mounting rail, proceed with the removal as follows:

1. Disconnect the power supply from those devices on which you have mounted the fieldbus coupler.
2. To remove the fieldbus coupler, press down the release eyebolt of the mounting rail adapter using a slot screwdriver (B) and remove it from the mounting rail (C).

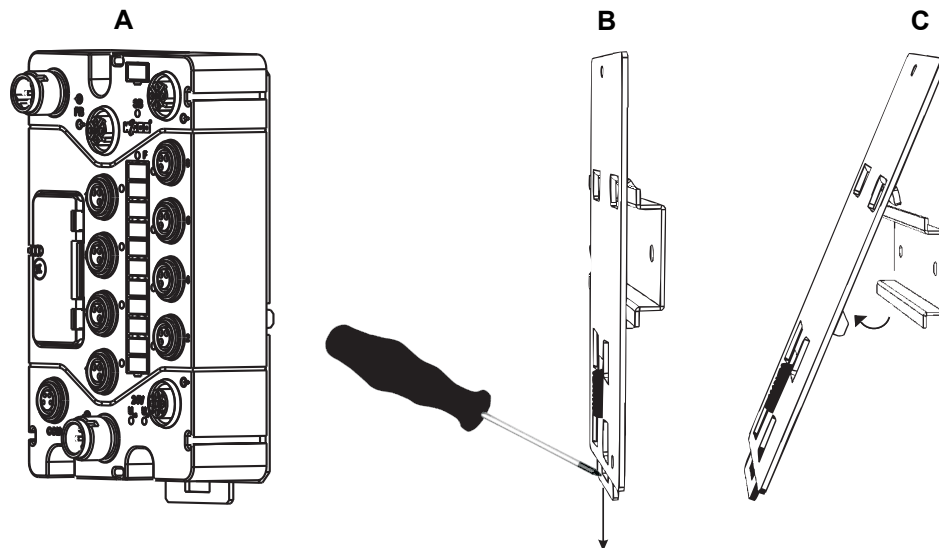


Figure 136: Removing the fieldbus coupler (with the mounting rail adapter) from the mounting rail

17.2.4 Removing the Fieldbus Coupler from the Profile Adapter

If the fieldbus coupler is mounted on a profile adapter, proceed with the removal as follows:

1. Disconnect the power supply from the device on which you have mounted the fieldbus coupler before attempting to remove.
2. Unscrew the screws on which the nuts are fastened and remove the fieldbus coupler from the profile rail of your system.
3. Unscrew the screws that connect the fieldbus coupler with the profile adapter.

17.2.5 Connecting a New Fieldbus Coupler

To connect the fieldbus coupler, proceed as described in Sections 7 and 8.

17.3 Disposal

Do not dispose of the 767 components in the household waste; observe the laws which apply to them. You can also contact a certified waste management company.

18 Appendix

18.1 WAGO-Specific Classes

The following table provides explanations of the column headers that can appear in the tables of the classes:

Table 148: Explanations of the column headers

Column heading	Description
Attribute ID	Integer value which is assigned to the respective attribute.
Access	Set: The attribute can be accessed using the Set_Attribute service (writing/modifying the attribute value). Important: If an attribute supports the Set_Attribute service, it can also be accessed with the Get_Attribute service. Get: The attribute can be accessed using the Get_Attribute services (reading the attribute value). Get_Attribute_All: Delivers the content of all attributes. Set_Attribute_Single: Modifies an attribute value.
NV	NV (non-volatile): The attribute is permanently saved in the fieldbus coupler. V (volatile): The attribute is not permanently saved in the fieldbus coupler.
Name	Identification of the attribute.
Data type	Identification of the CIP data type of the attribute.
Description	Brief description of the attribute.
Default value	Factory setting

The following table provides an overview of all WAGO-specific classes:

Table 149: WAGO-specific classes

Name	Class	Information
Coupler Configuration Object	64 _{hex}	See Section 18.1.1
Node Information Class	82 _{hex}	See Section 18.1.6
Discrete Input Point	65 _{hex}	See Section 18.1.2.1
Discrete Input Point Extended 1	69 _{hex}	See Section 18.1.2.2
Discrete Input Point Extended 2	6D _{hex}	See Section 18.1.2.3
Discrete Input Point Extended 3	71 _{hex}	See Section 18.1.2.4
Discrete Output Point	66 _{hex}	See Section 18.1.3.1

Table 149: WAGO-specific classes

Name	Class	Information
Discrete Output Point Extended 1	6A _{hex}	See Section 18.1.3.2
Discrete Output Point Extended 2	6E _{hex}	See Section 18.1.3.3
Discrete Output Point Extended 3	72 _{hex}	See Section 18.1.3.4
Analog Input Point	67 _{hex}	See Section 18.1.4.1
Analog Input Point Extended 1	6B _{hex}	See Section 18.1.4.2
Analog Input Point Extended 2	6F _{hex}	See Section 18.1.4.3
Analog Input Point Extended 3	73 _{hex}	See Section 18.1.4.4
Analog Output Point	68 _{hex}	See Section 18.1.5.1
Analog Output Point Extended 1	6C _{hex}	See Section 18.1.5.2
Analog Output Point Extended 2	70 _{hex}	See Section 18.1.5.3
Analog Output Point Extended 3	74 _{hex}	See Section 18.1.5.4
IOM Input Point Class	83 _{hex}	See Section 18.1.6
IOM Output Point Class	84 _{hex}	See Section 18.1.7
Input Fieldbus Variable USINT	A0 _{hex}	See Section 18.1.9.1
Input Fieldbus Variable USINT Extended 1	A1 _{hex}	See Section 18.1.9.2
Input Fieldbus Variable USINT Extended 2	A2 _{hex}	See Section 18.1.9.3
Input Fieldbus Variable UINT	A6 _{hex}	See Section 18.1.9.4
Input Fieldbus Variable UINT Extended 1	A7 _{hex}	See Section 18.1.9.5
Input Fieldbus Variable UDINT	AA _{hex}	See Section 18.1.9.6
Input Fieldbus Variable UDINT Offset UINT	AB _{hex}	See Section 18.1.9.7
Output Fieldbus Variable USINT	A3 _{hex}	See Section 18.1.10.1
Output Fieldbus Variable USINT Extended 1	A4 _{hex}	See Section 18.1.10.2
Output Fieldbus Variable USINT Extended 2	A5 _{hex}	See Section 18.1.10.3
Output Fieldbus Variable UINT	A8 _{hex}	See Section 18.1.10.4
Output Fieldbus Variable UINT Extended 1	A9 _{hex}	See Section 18.1.10.5
Output Fieldbus Variable UDINT	AC _{hex}	See Section 18.1.10.6
Output Fieldbus Variable UDINT Offset UINT	AD _{hex}	See Section 18.1.10.7

18.1.1 Bus Coupler Configuration Class Object (64_{hex})

The fieldbus coupler Configuration Class allows reading and configuration of some important fieldbus process parameters. The following listings explain in details all supported instances and attributes.

Class attribute

Table 150: "Bus coupler configuration class object" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	1 (0x0001)

Instance attribute 1

Table 151: "Bus coupler configuration class object" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
5	Get	V	Fieldbus process state	USINT	Fieldbus coupler status Bit 0: Error of process input data (0x01) Bit 1: Error of process output data (0x02) Bit 2: Loss of diagnostic information (0x04) Bit 3: Module diagnostics (0x08) Bit 7: Fieldbus error (0x80)	0 (0x00)
7	Get	NV	Configured length of analog outputs	UINT	Number of I/O bits of analog outputs	-
8	Get	NV	Configured length of analog inputs	UINT	Number of I/O bits of analog inputs	-
9	Get	NV	Configured length of digital outputs	UINT	Number of I/O bits of digital outputs	-
10	Get	NV	Configured length of digital inputs	UINT	Number of I/O bits of digital inputs	-
11	Get/Set	NV	Fieldbus fault reaction	USINT	Response to fieldbus error 1: Set all outputs to 0 2: No reaction	1 (0x01)
45	Get	V	Error code	USINT	Error code of I/O LED	0 = no error

Table 151: "Bus coupler configuration class object" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
46	Get	V	Error argument	USINT	Error argument of I/O LED	0 = no error
52	Get/Set	NV	S-BUS fault reaction	USINT	Response to S-BUS error 1: Set all inputs in the process image to 0 2: No reaction	1 (0x01)
53	Get	V	Maximum diagnostic data size	UINT	Maximum quantity of purely diagnostic data in bytes	-
54	Get	V	Diagnostic message	ARRAY OF USINT	Diagnostic consisting of diagnostic path and diagnostic data	-
55	Get/Set	V	Diagnostic message acknowledgement	USINT ARRAY	Diagnostic confirmation consisting of diagnostic path and diagnostic data	-
57	Get	V	Error group	USINT	Error group of the I/O LED	0 = no error
100	Get/Set	NV	Fieldbus input variable counter	UINT	Number of bytes for the fieldbus input variables that are to be added to assembly instances 101 through 103 and 113	0 (0x0000)
101	Get/Set	NV	Fieldbus output variable counter	UINT	Number of bytes for the fieldbus output variables that are to be added to assembly instances 104 through 109 and 112	0 (0x0000)
102	Get/Set	NV	PLC-only input variable counter	UINT	Number of bytes for the fieldbus input variables that are to be received via assembly instance 111	4 (0x0004)
103	Get/Set	NV	PLC-only start input variable counter	UINT	Start position of the fieldbus input variables that are to be received via assembly instance 111	0 (0x0000)

Table 151: "Bus coupler configuration class object" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
104	Get/Set	NV	PLC-only output variable counter	UINT	Number of bytes for the fieldbus output variables that are to be sent via assembly instance 110	4 (0x0004)
105	Get/Set	NV	PLC-only start output variable counter	UINT	Start position of the fieldbus output variables that are to be sent via assembly instance 110.	0 (0x0000)
120	Get/Set	NV	Header configuration originator to target	UINT	Indication of whether the RUN/IDLE header is used from the originator to the target. 0: Is used 1: Is not used	0 (0x0000)
121	Get/Set	NV	Header configuration target to originator	UINT	Indication of whether the RUN/IDLE header is used from the target to the originator. 0: Is used 1: Is not used	1 (0x0001)

Overview of services

Table 152: "Bus coupler configuration class object" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.2 Discrete Input Point Classes

There are 4 DIP classes allowing access to the node containing up to 1020 individual digital input points.

18.1.2.1 Discrete Input Point Class (65_{hex})

This class allows the reading of data of a particular digital input point. A digital input point (DIP) can be a part of digital input module or a digital input and output module. Each DIP can have a maximum width of 8 bits. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 153: "Discrete input point" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (digital input values 1 through 255)

Table 154: "Discrete input point" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Value	BYTE	Object value	-
2	Get	V	Value length	BYTE	Value length in bits	-

Overview of services

Table 155: "Discrete input point" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.2.2 Discrete Input Point Extended 1 (69_{hex})

The extension of the "discrete input point" class enables the reading of data from a 767 node that contains over 255 digital input points (DIPs). The instance scope of this class covers DIPs from 256 to 510 in the node. Each DIP can have a width of up to 8 bits. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 156: "Discrete input point extended 1" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (digital input values 256 through 510)

Table 157: "Discrete input point extended 1" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Value	BYTE	Object value	-
2	Get	V	Value length	BYTE	Value length in bits	-

Overview of services

Table 158: "Discrete input point extended 1" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.2.3 Discrete Input Point Extended 2 (6D_{hex})

The second extension of the "discrete input point" class enables the reading of data from a 767 node that contains over 510 DIPs. The instance scope of this class covers DIPs from 511 to 765. Each DIP can have a width of up to 8 bits. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 159: "Discrete input point extended 2" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (digital input values 511 through 765)

Table 160: "Discrete input point extended 2" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Value	BYTE	Object value	-
2	Get	V	Value length	BYTE	Value length in bits	-

Overview of services

Table 161: "Discrete input point extended 2" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.2.4 Discrete Input Point Extended 3 (71_{hex})

The third extension of the "discrete input point" class enables the reading of data from a 767 node that contains over 765 DIPs. The instance scope of this class covers DIPs from 766 to 1020. Each DIP can have a width of up to 8 bits. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 162: "Discrete input point extended 3" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (digital input values 766 through 1020)

Table 163: "Discrete input point extended 3" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Value	BYTE	Object value	-
2	Get	V	Value length	BYTE	Value length in bits	-

Overview of services

Table 164: "Discrete input point extended 3" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.3 Discrete Output Point Classes

There are 4 DOP classes allowing access to the node containing up to 1020 individual digital output points.

18.1.3.1 Discrete Output Point Class (66_{hex})

This class enables data exchange for a particular digital output point. A digital output point (DOP) can be a part of digital output module or a digital input and output module. Each DOP can have a width of up to 8 bits. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 165: "Discrete output point" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (digital output values 1 through 255)

Table 166: "Discrete output point" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Value	BYTE	Object value	-
2	Get	V	Value length	BYTE	Value length in bits	-

Overview of services

Table 167: "Discrete output point" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.3.2 Discrete Output Point Extended 1 (6A_{hex})

The first extension of the "discrete output point" class enables data exchange in a 767 node that contains over 255 DOPs. The instance scope of this class covers DOPs from 256 to 510. Each DOP can have a width of up to 8 bits. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 168: "Discrete output point extended 1" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (digital output values 256 through 510)

Table 169: "Discrete output point extended 1" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Value	BYTE	Object value	-
2	Get	V	Value length	BYTE	Value length in bits	-

Overview of services

Table 170: "Discrete output point extended 1" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.3.3 Discrete Output Point Extended 2 (6E_{hex})

The second extension of the "discrete output point" class enables data exchange in a 767 node that contains over 510 DOPs. The instance scope of this class covers DOPs from 511 to 765. Each DOP can have a width of up to 8 bits. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 171: "Discrete output point extended 2" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (digital output values 511 through 765)

Table 172: "Discrete output point extended 2" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Value	BYTE	Object value	-
2	Get	V	Value length	BYTE	Value length in bits	-

Overview of services

Table 173: "Discrete output point extended 2" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.3.4 Discrete Output Point Extended 3 (72_{hex})

The third extension of the "discrete output point" class enables data exchange in a 767 node that contains over 765 DOPs. The instance scope of this class covers DOPs from 766 to 1020. Each DOP can have a width of up to 8 bits. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 174: "Discrete output point extended 3" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (digital output values 766 through 1020)

Table 175: "Discrete output point extended 3" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Value	BYTE	Object value	-
2	Get	V	Value length	BYTE	Value length in bits	-

Overview of services

Table 176: "Discrete output point extended 3" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.4 Analog Input Point Classes

There are 4 AIP classes allowing access to the 767 node containing up to 1020 individual analog input points.

18.1.4.1 Analog Input Point Class (67_{hex})

This class enables the reading of data of a particular analog input point. An analog input point is part of an analog input module. Each AIP can contain up to 65535 bytes. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 177: "Analog input point" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (analog input values 1 through 255)

Table 178: "Analog input point" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Value	BYTE ARRAY	Object value	-
2	Get	V	Value length	BYTE	Value length in bytes	-
3	Get	V	Extended value length	WORD	Extended value range of values larger than 254 bytes	-

Overview of services

Table 179: "Analog input point" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.4.2 Analog Input Point Extended 1 (6B_{hex})

The first extension of the "analog input point" class enables the reading of data from a 767 node that contains over 255 AIPs. The instance scope of this class covers AIPs from 256 to 510. Each AIP can contain up to 65535 bytes. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 180: "Analog input point extended 1" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (analog input values 256 through 510)

Table 181: "Analog input point extended 1" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Value	BYTE ARRAY	Object value	-
2	Get	V	Value length	BYTE	Value length in bytes	-
3	Get	V	Extended value length	WORD	Extended value range of values larger than 254 bytes	-

Overview of services

Table 182: "Analog input point extended 1" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.4.3 Analog Input Point Extended 2 (6F_{hex})

The second extension of the "analog input point" class enables the reading of data from a 767 node that contains over 510 AIPs. The instance scope of this class covers AIPs from 511 to 765. Each AIP can contain up to 65535 bytes. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 183: "Analog input point extended 2" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (analog input values 511 through 765)

Table 184: "Analog input point extended 2" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Value	BYTE ARRAY	Object value	-
2	Get	V	Value length	BYTE	Value length in bytes	-
3	Get	V	Extended value length	WORD	Extended value range of values larger than 254 bytes	-

Overview of services

Table 185: "Analog input point extended 2" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.4.4 Analog Input Point Extended 3 (73_{hex})

The third extension of the "analog input point" class enables the reading of data from a 767 node that contains over 765 AIPs. The instance scope of this class covers AIPs from 766 to 1020. Each AIP can contain up to 65535 bytes. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 186: "Analog input point extended 3" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (analog input values 766 through 1020)

Table 187: "Analog input point extended 3" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Value	BYTE ARRAY	Object value	-
2	Get	V	Value length	BYTE	Value length in bytes	-
3	Get	V	Extended value length	WORD	Extended value range of values larger than 254 bytes	-

Overview of services

Table 188: "Analog input point extended 3" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.5 Analog Output Point Classes

There are four AOP classes allowing access to the node containing up to 1020 individual analog output points.

18.1.5.1 Analog Output Point (68_{hex})

This class enables the reading of data of a particular analog output point. An analog output point is part of an analog output module. Each AOP can contain up to 65535 bytes. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 189: "Analog output point" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (analog output values 1 through 255)

Table 190: "Analog output point" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Value	BYTE ARRAY	Object value	-
2	Get	V	Value length	BYTE	Value length in bytes	-
3	Get	V	Extended value length	WORD	Extended value range of values larger than 254 bytes	-

Overview of services

Table 191: "Analog output point" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.5.2 Analog Output Point Extended 1 (6C_{hex})

The first extension of the "analog output point" class enables the reading of data from a 767 node that contains over 255 AOPs. The instance scope of this class covers AOPs from 256 to 510. Each AOP can contain up to 65535 bytes. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 192: "Analog output point extended 1" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (analog output values 256 through 510)

Table 193: "Analog output point extended 1" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Value	BYTE ARRAY	Object value	-
2	Get	V	Value length	BYTE	Value length in bytes	-
3	Get	V	Extended value length	WORD	Extended value range of values larger than 254 bytes	-

Overview of services

Table 194: "Analog output point extended 1" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.5.3 Analog Output Point Extended 2 (70_{hex})

The second extension of the "analog output point" class enables the reading of data from a 767 node that contains over 510 AOPs. The instance scope of this class covers AOPs from 511 to 765. Each AOP can contain up to 65535 bytes. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 195: "Analog output point extended 2" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (analog output values 511 through 765)

Table 196: "Analog output point extended 2" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Value	BYTE ARRAY	Object value	-
2	Get	V	Value length	BYTE	Value length in bytes	-
3	Get	V	Extended value length	WORD	Extended value range of values larger than 254 bytes	-

Overview of services

Table 197: "Analog output point extended 2" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.5.4 Analog Output Point Extended 3 (74_{hex})

The third extension of the "analog output point" class enables the reading of data from a 767 node that contains over 765 AOPs. The instance scope of this class covers AOPs from 766 to 1020. Each AOP can contain up to 65535 bytes. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 198: "Analog output point extended 3" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	-

Instance attributes of instances 1 through 255 (analog output values 766 through 1020)

Table 199: "Analog output point extended 3" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Value	BYTE ARRAY	Object value	-
2	Get	V	Value length	BYTE	Value length in bytes	-
3	Get	V	Extended value length	WORD	Extended value range of values larger than 254 bytes	-

Overview of services

Table 200: "Analog output point extended 3" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.6 IOM Input Point Class (83_{hex})

This class provides information about an individual I/O modules. This includes the breadth of process information for the input points of the I/O module in bytes. In addition, this class also provides access to the process information of the module. The following tables provide an overview of the instances and attributes supported by WAGO:

Class Attributes

Table 201: "IOM Input Point Class" class: class attributes

Attribute ID	Access	Name	Data Type	Description	Default Value
1	Get	Revision	UINT	Version des Objektes	1 (0x0001)
2	Get	Max Instance	UINT	Max. Anzahl an Instanzen: 65	-

Instance Attributes of Instances 1 through 65 (I/O modul 1 through 65)

Attribute ID	Access	Name	Data Type	Description	Default Value
1	Get	V	Value	BYTE ARRAY	Objektwert
2	Get	V	Value Length	BYTE	Wertlänge in Byte



Note

The instances comply with the respective position of the I/O modules connected to the fieldbus coupler. Instance 1 refers to the integrated digital inputs or outputs of the fieldbus coupler. Instance 2 refers to the first I/O module connected to the fieldbus coupler, instance 3 to the second I/O module, etc. If an I/O module has no input point, then instance attribute 2 (Value Length) has the value 0.

Overview of Services

Table 202: "IOM Input Point Class" class: services

Service Code	Service Available		Service Name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Liefert den Inhalt des entsprechenden Attributes

18.1.7 IOM Output Point Class (84_{hex})

This class provides information about an individual I/O modules. This includes the breadth of process information for the output points of the I/O module in bytes. In addition, this class also provides access to the process information of the module. The following tables provide an overview of the instances and attributes supported by WAGO:

Class Attributes

Table 203: "IOM Output Point Class" class: class attributes

Attribute ID	Access	Name	Data Type	Description	Default Value
1	Get	Revision	UINT	Version des Objektes	1 (0x0001)
2	Get	Max Instance	UINT	Max. Anzahl an Instanzen: 65	-

Instance Attributes of Instances 1 through 65 (I/O modul 1 through 65)

Attribute ID	Access	Name	Data Type	Description	Default Value
1	Get/Set	V	Value	BYTE ARRAY	Objektwert
2	Get	V	Value Length	BYTE	Wertlänge in Byte



Note

The instances comply with the respective position of the I/O modules connected to the fieldbus coupler. Instance 1 refers to the integrated digital inputs or outputs of the fieldbus coupler. Instance 2 refers to the first I/O module connected to the fieldbus coupler, instance 3 to the second I/O module, etc. If an I/O module has no output point, then instance attribute 2 (Value Length) has the value 0.

Overview of Services

Table 204: "IOM Output Point Class" class: services

Service Code	Service Available		Service Name	Description
	Class	Instance		
0E hex	Yes	Ja	Get_Attribute_Single	Liefert den Inhalt des entsprechenden Attributes
10 hex	No	Ja	Set_Attribute_Single	Modifiziert einen Attributwert

18.1.8 Node Information Class (82_{hex})

This class provides information about an individual I/O module. Among the more important information is the item number. Additional information about the I/O module can be found in instance attribute 2. From this instance attribute, you can identify whether the I/O module is physical or virtual and whether the MODBUS process image, EtherNet/IP or an IEC 61131 application has been assigned.

Class attribute

Table 205: "Node information" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances: 65	-

Instance attributes of instances 1 through 255 (I/O modules 1 through 65)

Table 206: "Node information" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Order number	SHORT_STRING	Item number	-
2	Get	V	Additional info	UDINT	*	-

Overview of services

Table 207: "Node information" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

* **Bit 0:** Additional module information **0** = Physical I/O module, **1** = Virtual I/O module

Bit 1: The process image is assigned to EtherNet/IP **0** = No, **1** = Yes

18.1.9 Input Fieldbus Variables Classes

There are 512 bytes provided for input fieldbus variables. For full access to this memory area, byte-based, word-based, and double-word-based classes are provided. Input variables can be both read and written.

18.1.9.1 Input Fieldbus Variable USINT (A0_{hex})

This class enables byte-by-byte access to the first 255 bytes of the memory area of the input fieldbus variables. Each byte in this memory area is represented by a particular instance of this class. The following listings explain in details all supported instances and attributes.

Class attribute

Table 208: "Input Fieldbus Variable USINT" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	255 (0x0FF)

Instance attribute 1 for instances 1 through 255 (input variables 1 through 255)

Table 209: "Input Fieldbus Variable USINT" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Input variable	BYTE	Fieldbus input variable of PLC	-

Overview of services

Table 210: "Input Fieldbus Variable USINT" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.9.2 Input Fieldbus Variable USINT Extended 1 (A1_{hex})

The first extension of the "Input Fieldbus Variable USINT" class enables byte-by-byte access to bytes 256 through 510 of the memory area of the input fieldbus variables. Each byte in this memory area is represented by a particular instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 211: "Input Fieldbus Variable USINT Extended 1" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	255 (0x0FF)

Instance attribute 1 for instances 1 through 255 (input variables 256 through 510)

Table 212: "Input Fieldbus Variable USINT Extended 1" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Input variable	BYTE	Fieldbus input variable of PLC	-

Overview of services

Table 213: "Input Fieldbus Variable USINT Extended 1" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.9.3 Input Fieldbus Variable USINT Extended 2 (A2_{hex})

The second extension of the "Input Fieldbus Variable USINT" class enables byte-by-byte access to bytes 511 and 512 of the memory area of the input fieldbus variables. Both bytes in this memory area are represented by a particular instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 214: "Input Fieldbus Variable USINT Extended 2" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	2 (0x0002)

Instance attribute 1 for instances 1 and 2 (input variables 511 and 512)

Table 215: "Input Fieldbus Variable USINT Extended 2" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Input variable	BYTE	Fieldbus input variable of PLC	-

Overview of services

Table 216: "Input Fieldbus Variable USINT Extended 2" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.9.4 Input Fieldbus Variable UINT (A6_{hex})

This class enables word-by-word access to the first 255 bytes of the memory area of the input fieldbus variables. Each word in this memory area is represented by a particular instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 217: "Input Fieldbus Variable UINT" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	255 (0x0FF)

Instance attribute 1 for instances 1 through 255 (input variables 1 through 255)

Table 218: "Input Fieldbus Variable UINT" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Input variable	WORD	Fieldbus input variable of PLC	-

Overview of services

Table 219: "Input Fieldbus Variable UINT" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.9.5 Input Fieldbus Variable UINT Extended 1 (A7_{hex})

The first extension of the "Input Fieldbus Variable USINT" class enables byte-by-byte access to bytes 256 through 510 of the memory area of the fieldbus input variables. The word 256 in this memory area is represented by only one instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 220: "Input Fieldbus Variable UINT Extended 1" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	1 (0x0001)

Instance attribute 1 for instance 1 (input variable 256)

Table 221: "Input Fieldbus Variable UINT Extended 1" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Input variable	WORD	Fieldbus input variable of PLC	-

Overview of services

Table 222: "Input Fieldbus Variable UINT Extended 1" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.9.6 Input Fieldbus Variable UDINT (AA_{hex})

This class enables double-word access to the first 128 double-words of the memory area of the input fieldbus variables. Each double-word in this memory is represented by a particular instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 223: "Input Fieldbus Variable UDINT" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	128 (0x080)

Instance attribute 1 for instances 1 through 128 (input variables 1 through 128)

Table 224: "Input Fieldbus Variable UDINT" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Input variable	DWORD	Fieldbus input variable of PLC	-

Overview of services

Table 225: "Input Fieldbus Variable UDINT" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

18.1.9.7 Input Fieldbus Variable UDINT Offset UINT (AB_{hex})

This class also enables double-word-based access to 128 double-words of the memory area of the input fieldbus variables. However, unlike the "Input Fieldbus Variable UDINT" class, all instances of this class are word-oriented and thus have an offset of 2 bytes.

Class attribute

Table 226: "Input Fieldbus Variable UDINT Offset UINT" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	128 (0x080)

Instance attribute 1 for instances 1 through 128 (input variables 1 through 128)

Table 227: "Input Fieldbus Variable UDINT Offset UINT" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get/Set	V	Input variable	DWORD	Fieldbus input variable of PLC	-

Overview of services

Table 228: "Input Fieldbus Variable UDINT Offset UINT" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

Access to the last instance is limited. Through the reading of this instance, two upper bytes are always 0x00. Through the writing, the two upper bytes are omitted regardless of the value that is to be written. Thus, in actuality, access to the last instance is merely a word-by-word access.

18.1.10 Output Fieldbus Variables Classes

There are 512 bytes provided for output fieldbus variables. For full access to this memory area, byte-based, word-based, and double-word-based classes are provided. Output variables can only be read out.

18.1.10.1 Output Fieldbus Variable USINT (A3_{hex})

This class enables byte-by-byte read access to the first 255 bytes of the memory area of the fieldbus output variables. Each byte in this memory area is represented by a particular instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 229: "Output Fieldbus Variable USINT" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	255 (0x0FF)

Instance attribute 1 for instances 1 through 255 (output variables 1 through 255)

Table 230: "Output Fieldbus Variable USINT" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Input variable	BYTE	Fieldbus input variable of PLC	-

Overview of services

Table 231: "Output Fieldbus Variable USINT" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.10.2 Output Fieldbus Variable USINT Extended 1 (A4_{hex})

The first extension of the "Output Fieldbus Variable USINT" class enables byte-by-byte read access to bytes 256 through 510 of the memory area of the output fieldbus variables. Each byte in this memory area is represented by a particular instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 232: "Output Fieldbus Variable USINT Extended 1" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	255 (0x0FF)

Instance attribute 1 for instances 1 through 255 (output variables 256 through 510)

Table 233: "Output Fieldbus Variable USINT Extended 1" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Input variable	BYTE	Fieldbus input variable of PLC	-

Overview of services

Table 234: "Output Fieldbus Variable USINT Extended 1" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.10.3 Output Fieldbus Variable USINT Extended 2 (A5_{hex})

The second extension of the "Output Fieldbus Variable USINT" class enables byte-by-byte read access to bytes 511 and 512 of the memory area of the output fieldbus variables. Both bytes in this memory area are represented by a particular instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 235: "Fieldbus Output Variable Byte 2 Class (A5_{hex})" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	2 (0x0002)

Instance attribute 1 for instances 1 and 2 (output variables 511 and 512)

Table 236: "Fieldbus Output Variable Byte 2 Class (A5_{hex})" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Input variable	BYTE	Fieldbus input variable of PLC	-

Overview of services

Table 237: "Fieldbus Output Variable Byte 2 Class (A5_{hex})" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.10.4 Output Fieldbus Variable UINT (A8_{hex})

This class enables word-by-word read access to the first 255 bytes of the memory area of the output fieldbus variables. Each word in this memory area is represented by a particular instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 238: "Fieldbus Output Variable Word Class (A8_{hex})" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	255 (0xFF)

Instance attribute 1 for instances 1 through 255 (output variables 1 through 255)

Table 239: "Fieldbus Output Variable Word Class (A8_{hex})" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Input variable	WORD	Fieldbus input variable of PLC	-

Overview of services

Table 240: "Fieldbus Output Variable Word Class (A8_{hex})" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.10.5 Output Fieldbus Variable UINT Extended 1 (A9_{hex})

The first extension of the "Output Fieldbus Variable UINT" class enables word-by-word read access to word 256 of the memory area of the output fieldbus variables. The word 256 in this memory area is represented by only one instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 241: "Fieldbus Output Variable Word 1 Class (A9_{hex})" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	1 (0x0001)

Instance attribute 1 for instance 1 (output variable 256)

Table 242: "Fieldbus Output Variable Word 1 Class (A9_{hex})" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Input variable	WORD	Fieldbus input variable of PLC	-

Overview of services

Table 243: "Fieldbus Output Variable Word 1 Class (A9_{hex})" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.10.6 Output Fieldbus Variable UDINT (AC_{hex})

This class enables double-word read access to the 128 double-words of the memory area of the output fieldbus variables. Each double-word in this memory is represented by a particular instance of this class. The following tables provide an overview of the instances and attributes supported by WAGO:

Class attribute

Table 244: "Fieldbus Output Variable Double-Word Class (AC_{hex})" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	128 (0x080)

Instance attribute 1 for instances 1 through 128 (output variables 1 through 128)

Table 245: "Fieldbus Output Variable Double-Word Class (AC_{hex})" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Input variable	DWORD	Fieldbus input variable of PLC	-

Overview of services

Table 246: "Fieldbus Output Variable Double-Word Class (AC_{hex})" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.1.10.7 Output Fieldbus Variable UDINT Offset UINT (AD_{hex})

This class also enables double-word-based access to 128 double-words of the memory area of the output fieldbus variables. However, unlike the "Output Fieldbus Variable UDINT" class, all instances of this class are word-oriented and thus have an offset of 2 bytes.

Class attribute

Table 247: "Fieldbus Output Variable Double-Word 1 Class (AD_{hex})" class: class attribute

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Version of object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum number of instances	128 (0x080)

Instance attribute 1 for instance 1 (output variable 128)

Table 248: "Fieldbus Output Variable Double-Word 1 Class (AD_{hex})" class: instance attribute

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Input variable	DWORD	Fieldbus input variable of PLC	-

Overview of services

Table 249: "Fieldbus Output Variable Double-Word 1 Class (AD_{hex})" class: services

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Delivers content of respective attribute

18.2 Available Libraries

After installing WAGO CoDeSys, the following libraries are available to you for the fieldbus coupler:

Table 250: Libraries

Name	Description
FtpClient	-
Config	Access to firmware version, version of FW components, WAGO_Object_Lib
Errors	-

18.3 Accessories

Here you will find a list of the most important accessory components that you need to begin operating the fieldbus coupler. The mounting rails, spacers, etc., as well as the entire accessories program from the AUTOMATION field, can be found in the internet as usual or in the "Components for Automation" catalog.

18.3.1 S-BUS cable, assembled on both ends

Table 251: S-BUS cable, assembled on both ends, B coded

Item	Length	Item number	PU
M12 socket, M12 plug, straight	0.2 m	756-1305/0060-0002	1
M12 socket, M12 plug, straight	0.3 m	756-1305/0060-0003	
M12 socket, M12 plug, straight	0.5 m	756-1305/0060-0005	
M12 socket, M12 plug, straight	1.0 m	756-1305/0060-0010	
M12 socket, M12 plug, straight	2.0 m	756-1305/0060-0020	
M12 socket, M12 plug, straight	5.0 m	756-1305/0060-0050	
M12 socket, M12 plug, straight	10.0 m	756-1305/0060-0100	
M12 socket, M12 plug, straight	20.0 m	756-1305/0060-0200	
M12 socket, M12 plug, straight	50.0 m	756-1305/0060-0500	
M12 socket, M12 plug, angular	0.2 m	756-1306/0060-0002	
M12 socket, M12 plug, angular	0.3 m	756-1306/0060-0003	
M12 socket, M12 plug, angular	0.5 m	756-1306/0060-0005	
M12 socket, M12 plug, angular	1.0 m	756-1306/0060-0010	
M12 socket, M12 plug, angular	2.0 m	756-1306/0060-0020	
M12 socket, M12 plug, angular	5.0 m	756-1306/0060-0050	
M12 socket, M12 plug, angular	10.0 m	756-1306/0060-0100	
M12 socket, M12 plug, angular	20.0 m	756-1306/0060-0200	
M12 socket, M12 plug, angular	50.0 m	756-1306/0060-0500	

18.3.2 S-BUS Terminator and USB Cable

Table 252: S-BUS terminator (B coded) and USB cable

Column heading	Item number	PU
M12 S-BUS terminating plug, B coded	756-9409/0060-0000	1
USB communication cable	756-4101/0042-0030	

18.3.3 Supply Cable Assembled on Both Ends

Table 253: Supply cable, assembled on both ends, A coded

Item	Length	Item number	PU
M12 socket, M12 plug, straight	0.2 m	756-3105/0040-0002	1
M12 socket, M12 plug, straight	0.3 m	756-3105/0040-0003	
M12 socket, M12 plug, straight	0.5 m	756-3105/0040-0005	
M12 socket, M12 plug, straight	1.0 m	756-3105/0040-0010	
M12 socket, M12 plug, straight	2.0 m	756-3105/0040-0020	
M12 socket, M12 plug, straight	5.0 m	756-3105/0040-0050	
M12 socket, M12 plug, straight	10.0 m	756-3105/0040-0100	
M12 socket, M12 plug, straight	20.0 m	756-3105/0040-0200	
M12 socket, M12 plug, angular	0.2 m	756-3106/0040-0002	
M12 socket, M12 plug, angular	0.3 m	756-3106/0040-0003	
M12 socket, M12 plug, angular	0.5 m	756-3106/0040-0005	
M12 socket, M12 plug, angular	1.0 m	756-3106/0040-0010	
M12 socket, M12 plug, angular	2.0 m	756-3106/0040-0020	
M12 socket, M12 plug, angular	5.0 m	756-3106/0040-0050	
M12 socket, M12 plug, angular	10.0 m	756-3106/0040-0100	
M12 socket, M12 plug, angular	20.0 m	756-3106/0040-0200	

18.3.4 Ethernet Cable Assembled on One End

Table 254: Ethernet cable, assembled on one end, D coded

Item	Length	Item number	PU
M12 plug, straight, open-ended	2 m	756-1201/0060-0020	1
M12 plug, straight, open-ended	5 m	756-1201/0060-0050	
M12 plug, straight, open-ended	10 m	756-1201/0060-0100	
M12 plug, straight, open-ended	20 m	756-1201/0060-0200	
M12 plug, angular, open-ended	2 m	756-1202/0060-0020	
M12 plug, angular, open-ended	5 m	756-1202/0060-0050	
M12 plug, angular, open-ended	10 m	756-1202/0060-0100	
M12 plug, angular, open-ended	20 m	756-1202/0060-0200	

18.3.5 Ethernet Cable Assembled on Both Ends

Table 255: Ethernet cable, assembled on both ends, D coded

Item	Length	Item number	PU
M12 plug, straight, M12 plug, straight	2 m	756-1203/0060-0020	1
M12 plug, straight, M12 plug, straight	5 m	756-1203/0060-0050	
M12 plug, straight, M12 plug, straight	10 m	756-1203/0060-0100	
M12 plug, straight, M12 plug, straight	20 m	756-1203/0060-0200	
M12 plug, angular M12 plug, angular	2 m	756-1204/0060-0020	
M12 plug, angular M12 plug, angular	5 m	756-1204/0060-0050	
M12 plug, angular M12 plug, angular	10 m	756-1204/0060-0100	
M12 plug, angular M12 plug, angular	20 m	756-1204/0060-0200	

18.3.6 Ethernet Accessories

Table 256: Ethernet accessories

Column heading	Item number	PU
M12 plug, straight, D coded Can be preassembled; Spring clamp technology	756-9501/0060-0000	1
M12 plug, angular, D coded Can be preassembled; Spring clamp technology	756-9501/0040-0000	
Adapter, M12 socket, D coded/RJ-45 socket	756-9503/0040-0000	
Ethernet plug RJ 45, IP 20	750-975	

18.3.7 Carrier Rail Adapters and Profile Adapters

Table 257: Mounting rail adapters and profile adapters

Column heading	Item number	PU
Mounting rail adapter	On request	1
Profile adapter	On request	

18.3.8 Protective Caps

Table 258: Protective caps for unassigned connections sockets

Column heading	Item number	PU
M8 protective caps for connection sockets	756-8101	1
M12 protective caps for connection socket	756-8102	
M12 protective caps for connection plugs	755-809	
M23 protective caps for connection plugs	755-837	

List of Figures

Figure 1: Identification of connections	39
Figure 2: Identification of possibilities for marking and fastening	40
Figure 3: Identification of LEDs	41
Figure 4: DIP switch. All switches are set on "Off" at delivery.....	43
Figure 5: Marking and symbols.....	44
Figure 6: Label on fieldbus coupler	45
Figure 6: Block diagram.....	46
Figure 7: Dimensions of fieldbus coupler in mm.....	47
Figure 8: Mounting the fieldbus coupler on the grounded system.....	52
Figure 9: Fastening the fieldbus coupler to the carrier rail adapter.....	53
Figure 10: Mounting the carrier rail adapter	54
Figure 11: Fastening the fieldbus coupler to the profile adapter.....	55
Figure 12: Replacing the marking spaces.....	57
Figure 13: Attaching a spacer.....	58
Figure 14: Attaching another 767 component to the fieldbus coupler	59
Figure 15: The fieldbus coupler is connected to the Ethernet network.....	63
Figure 16: Example setup of a mixed topology with four fieldbus couplers	64
Figure 17: Example setup of a mixed topology with four fieldbus couplers	65
Figure 18: S-BUS is connected to one fieldbus coupler and two I/O modules.....	67
Figure 19: Supply cable connected with fieldbus coupler and I/O modules.....	69
Figure 20: Digital inputs.....	71
Figure 21: USB connection	73
Figure 22: Example of a MAC address on the adhesive label on the left side of the fieldbus coupler.....	77
Figure 23: Configuration file.....	78
Figure 24: Dialog window of the WAGO BootP server with messages	79
Figure 25: DIP set for the host ID 100	81
Figure 26: Example of a fieldbus node function test.....	83
Figure 27: "Enter network password" dialog box	86
Figure 28: "Information" view	87
Figure 29: "TCP/IP" view	89
Figure 30: "Port" view.....	91
Figure 30: "Watchdog" view	92
Figure 30: "Watchdog" view 2	94
Figure 32: "Clock" view	95
Figure 33: "Ethernet" view	96
Figure 34: "Users" view	98
Figure 35: "Administration" view	99
Figure 36: Connection setup using FTB in the DOS console	101
Figure 37: "Remove Hardware" icon in the taskbar.....	162
Figure 38: "Remove or Eject Hardware Components" dialog box	162
Figure 39: Edit Gateway.cfg	163
Figure 40: "Gateway.cfg" configuration file.....	163
Figure 41: "New Project" dialog box	165
Figure 42: "Add Object" dialog box.....	166
Figure 43: "Add Object" dialog box with the "Library Manager" selection.....	167
Figure 44: "Library Manager" view	167
Figure 45: "Add Library" dialog box with the "Standard" selection.....	168

Figure 46: "Library Manager" view with the "Standard 3.0.0.0" library.....	168
Figure 47: Creating a control program.....	169
Figure 48: "Add Object" dialog box with the "POU" selection.....	170
Figure 49: Creating a task management object.....	171
Figure 50: "Add Object" dialog box with the "Task Configuration" selection...	172
Figure 51: Creating a task.....	173
Figure 52: "Add Object" dialog box.....	174
Figure 53: Assignment between POU and task.....	175
Figure 54: "Input Assistance" dialog box.....	175
Figure 55: System tray on your desktop.....	176
Figure 56: "Communication Settings" tab.....	176
Figure 57: Loading a PLC program.....	177
Figure 58: Confirmation prompt.....	178
Figure 59: Adding I/O modules.....	179
Figure 60: "Attach Device" dialog box.....	179
Figure 61: Selection of I/O modules in "Attach Device" dialog box.....	180
Figure 62: Link variables used in the program using inputs and outputs.....	181
Figure 63: Define variables used in the program using inputs and outputs.....	182
Figure 64: Adding device objects.....	183
Figure 65: "Add Device" dialog box.....	184
Figure 66: Inserted MODBUS input register.....	184
Figure 67: Assigning a MODBUS address.....	185
Figure 68: Link variables used in the program using inputs and outputs.....	186
Figure 69: Define variables used in the program using inputs and outputs.....	187
Figure 70: Adding Device.....	188
Figure 71: "Add Object" dialog box 1.....	188
Figure 72: "Add Object" dialog box 2.....	189
Figure 73: Added EtherNet/IP object.....	189
Figure 74: Added EtherNet/IP object with 0 bit offset.....	190
Figure 75: 1. Selected 32-bit object with bit offset of 0.....	191
Figure 76: 2. Selected 16-bit object with bit offset of 32.....	192
Figure 77: Link variables used in the program using inputs and outputs.....	193
Figure 78: Define variables used in the program using inputs and outputs.....	194
Figure 79: Creating a boot project.....	197
Figure 80: CoDeSys file explorer.....	198
Figure 81: Run/Stop switch (10).....	198
Figure 82: LEDs which display status messages.....	201
Figure 83: LEDs which display operational messages.....	203
Figure 84: Indicator of blink codes via CS LED.....	205
Figure 85: Detailed display of the blink code in the web-based management....	219
Figure 86: Example of the blink code display under the "Blink code" parameter in WAGOframe.....	219
Figure 87: Start screen of WAGOframe CD.....	221
Figure 88: Window for product selection 1.....	221
Figure 89: Window for product selection 2.....	222
Figure 90: WAGOframe logo.....	223
Figure 91: "Device Selection Wizard".....	223
Figure 92: "Query WAGOframe" dialog box.....	223
Figure 93: "Query WAGOframe" dialog box.....	224
Figure 94: View of device catalog.....	224

Figure 95: Adding the communication DTM.....	225
Figure 96: Selecting the communication DTM.....	226
Figure 97: The DTM for the interface configuration	227
Figure 98: Adding a coupler.....	228
Figure 99: Adding the coupler.....	228
Figure 100: Adding the I/O modules.....	229
Figure 101: Selecting an I/O module	229
Figure 102: Two added I/O modules.....	229
Figure 103: Opening the configuration interface (offline).....	230
Figure 104: Setting up a connection to the fieldbus coupler	232
Figure 105: Opening the configuration interface (online).....	233
Figure 106: Opening the "List of Bus Nodes" window.....	235
Figure 107: Assigning new bus addresses for the I/O modules	236
Figure 108: Opening the "Assign Owner" window.....	237
Figure 109: "Assign Owner" window	237
Figure 110: Selecting a new user for the selected I/O modules	238
Figure 111: Warning	238
Figure 112: Runtime system chosen as new user for the selected I/O modules ..	239
Figure 113: List of connected 767 components	240
Figure 114: Service page.....	241
Figure 115: User administration.....	242
Figure 116: Prompt to reset passwords	242
Figure 117: File system	243
Figure 118: Setting up the network (on I/O modules connected to the fieldbus coupler)	244
Figure 119: Components connected to the 767 node	244
Figure 120: Searching for connected I/O modules using the life list.....	245
Figure 121: Connected I/O modules of the selected fieldbus coupler	246
Figure 122: Adding the DTM system update.....	248
Figure 123: DTM system update.....	248
Figure 124: Go online to 767 node.....	249
Figure 125: Package management 1.....	250
Figure 126: Package management 2.....	251
Figure 127: System update 1	252
Figure 128: System update 2	253
Figure 129: System update 3	254
Figure 130: Example of an opened DTM with the available parameters.....	256
Figure 131: System parameter handling 1	259
Figure 132: System parameter handling 2.....	260
Figure 133: Example of the diagnostic overview of a module (information may differ from the actual module).....	267
Figure 134: Removing the fieldbus coupler (with the mounting rail adapter) from the mounting rail	271

List of Tables

Table 1: Default settings of DIP switch	43
Table 2: Explanation of DIP switch	43
Table 1: Description of manufacturing number	45
Table 3: Technical data	48
Table 4: Ethernet connection assignments	62
Table 5: S-BUS connection assignment	66
Table 6: Supply connection assignment	68
Table 7: Connection assignment of digital inputs	70
Table 8: USB connection assignment	72
Table 9: Explanations of the configuration line	78
Table 10: Default settings of DIP switch	80
Table 11: Example of access configuration for MODBUS and EtherNet/IP	85
Table 12: Description of the "Information" view parameters	87
Table 13: Description of the "TCP/IP" view parameters	90
Table 14: Description of the "Port" view parameters	91
Table 15: Description of the "Watchdog" view parameters	93
Table 23: Description of the "Watchdog" view parameters	94
Table 16: Description of the "Clock" view parameters	95
Table 17: Description of the "Ethernet" view parameters	97
Table 18: Summary of file system partitions	100
Table 19: User administration	100
Table 20: CIP classes	116
Table 21: Explanation of table headers	116
Table 22: "Identity object" class: class attribute	118
Table 23: "Identity object" class: instance attribute	118
Table 24: "Identity object" class: services	119
Table 25: "Message router object" class: class attribute	120
Table 26: "Message router object" class: instance attribute	121
Table 27: "Message router object" class: services	121
Table 28 "Assembly object" class: class attribute	122
Table 29: "Assembly object" class: instance attribute 101 (65 hex)	122
Table 30: "Assembly object" class: instance attribute 102 (66 hex)	123
Table 31: "Assembly object" class: instance attribute 103 (67 hex)	123
Table 32: "Assembly object" class: instance attribute 104 (68 hex)	124
Table 33: "Assembly object" class: instance attribute 105 (69 hex)	124
Table 34: "Assembly object" class: instance attribute 106 (6A _{hex})	125
Table 35: "Assembly object" class: instance attribute 107 (6B _{hex})	125
Table 36: "Assembly object" class: instance attribute 108 (6C _{hex})	126
Table 37: "Assembly object" class: instance attribute 109 (6D _{hex})	126
Table 38: "Assembly object" class: instance attribute 110 (6E _{hex})	126
Table 39: "Assembly object" class: instance attribute 111 (6F _{hex})	127
Table 40: "Assembly object" class: instance attribute 112 (70 _{hex})	127
Table 41: "Assembly object" class: instance attribute 113 (71 _{hex})	128
Table 42: "Assembly object" class; services	128
Table 43: "Port class object" class: class attribute	130
Table 44: "Port class object" class: instance attribute	130
Table 45: "Port class object" class: services	131
Table 46: "TCP/IP interface object" class: class attributes	132

Table 47: "TCP/IP interface object" class: instance attributes.....	132
Table 48: "TCP/IP interface object" class: services.....	133
Table 49: "Ethernet link object" class: class attribute.....	134
Table 50: "Ethernet link object" class: instance attribute 1.....	134
Table 51: "Ethernet link object" class: instance attribute 2.....	136
Table 52: "Ethernet link object" class: instance attribute 3.....	137
Table 53: "Ethernet link object" class: services.....	138
Table 54: WAGO-specific classes.....	139
Table 55: MODBUS function codes (FC).....	144
Tabelle 56: Read analog input terminals using FC3, FC4, FC23.....	145
Tabelle 57: Schreiben von Analogausgangsklemmen mittels FC6, FC16, FC22, FC23.....	146
Tabelle 58: Read digital input terminals using FC1, FC2.....	147
Tabelle 59: Write digital output terminal using FC5, FC15.....	148
Table 60: Configuration register.....	149
Table 61: Register address 0x1000.....	151
Table 62: Register address 0x1001.....	151
Table 63: Register address 0x1002.....	151
Table 64: Register address 0x1003.....	152
Table 65: Register address 0x1004.....	152
Table 66: Register address 0x1005.....	152
Table 67: Register address 0x1006.....	152
Table 68: Register address 0x1007.....	153
Table 69: Register address 0x1008.....	153
Table 70: Register address 0x1009.....	153
Table 71: Register address 0x100A.....	153
Table 72: Register address 0x100B.....	153
Table 73: Register address 0x1020.....	154
Table 74: Register address 0x1021.....	154
Table 75: Register address 0x1022.....	154
Table 76: Register address 0x1023.....	154
Table 77: Register address 0x1024.....	154
Table 78: Register address 0x1025.....	154
Table 79: Register address 0x1028.....	155
Table 80: Register address 0x1029.....	155
Table 81: Register address 0x102A.....	155
Table 82: Register address 0x1030.....	155
Table 83: Register address 0x1031.....	155
Table 84: Register address 0x2030.....	156
Table 85: Register address 0x2040.....	156
Table 86: Register address 0x2041.....	156
Table 87: Register address 0x2042.....	156
Table 88: Register address 0x2043.....	156
Table 89: Register address 0x2010.....	157
Table 90: Register address 0x2011.....	157
Table 91: Register address 0x2012.....	157
Table 92: Register address 0x2013.....	157
Table 93: Register address 0x2014.....	158
Table 94: Register address 0x2020.....	158
Table 95: Register address 0x2021.....	158

Table 96: Register address 0x2022	158
Table 97: Register address 0x2023	158
Table 98: Register address 0x2000	159
Table 99: Register address 0x2001	159
Table 100: Register address 0x2002	159
Table 101: Register address 0x2003	159
Table 102: Register address 0x2004	159
Table 103: Register address 0x2005	159
Table 104: Register address 0x2006	160
Table 105: Register address 0x2007	160
Table 106: Register address 0x2008	160
Table 107: Notes on CoDeSys functions	199
Table 108: Ethernet status messages	201
Table 109: Operating messages of the fieldbus coupler.....	203
Table 110: Overview of the blink codes	206
Table 111: List of error groups.....	208
Table 112: Group number 1: S-BUS error	209
Table 113: Group number 2: S-BUS warnings	210
Table 114: Group number 3: RTS error	211
Table 115: Group number 6: general internal hardware warnings.....	213
Table 116: Group number 7: general software errors	214
Table 117: Group number 8: general software warnings.....	215
Table 118: Group number 11: fieldbus-specific software error.....	216
Table 119: Group number 12: fieldbus-specific software warnings.....	217
Table 120: Group number 13: firmware loader error.....	217
Table 121: Group number 14: error with firmware download.....	218
Table 122: Service page	241
Table 123: User administration	242
Table 124: File system	243
Table 125: Buttons	253
Table 126: DTM buttons	256
Table 127: Information about the fieldbus coupler	257
Table 128: Time settings	257
Table 61: S-BUS-Master	258
Table 129: Display of errors.....	258
Table 130: Overview of adjustable parameters for "System Parameter Handling"	260
Table 131: Overview of system event history.....	261
Table 132: Overview of diagnostics history.....	262
Table 133: Parameters for Ethernet configuration	263
Table 134: Parameters for configuring the Ethernet IPV4.....	263
Table 135: Parameters for configuring the Ethernet IP.....	264
Table 136: Parameters for configuring the Ethernet MODBUS	265
Table 137: Parameters for configuring the SNTP	266
Table 138: Parameters for configuring the FTP server	266
Table 139: Parameters for configuring the WEB server	266
Table 140: Diagnostics setup	267
Table 141: Information about existing module diagnostics	267
Table 142: Overview of adjustable parameters for the digital inputs	268
Table 143: Overview of parameters for the entire fieldbus coupler.....	269

Table 144: Overview of adjustable parameters for the field supply	269
Table 145: Explanations of the column headers	273
Table 146: WAGO-specific classes	273
Table 147: "Bus coupler configuration class object" class: class attribute	275
Table 148: "Bus coupler configuration class object" class: instance attribute	275
Table 149: "Bus coupler configuration class object" class: services	277
Table 150: "Discrete input point" class: class attribute	278
Table 151: "Discrete input point" class: instance attribute	278
Table 152: "Discrete input point" class: services	278
Table 153: "Discrete input point extended 1" class: class attribute	279
Table 154: "Discrete input point extended 1" class: instance attribute	279
Table 155: "Discrete input point extended 1" class: services	279
Table 156: "Discrete input point extended 2" class: class attribute	280
Table 157: "Discrete input point extended 2" class: instance attribute	280
Table 158: "Discrete input point extended 2" class: services	280
Table 159: "Discrete input point extended 3" class: class attribute	281
Table 160: "Discrete input point extended 3" class: instance attribute	281
Table 161: "Discrete input point extended 3" class: services	281
Table 162: "Discrete output point" class: class attribute	282
Table 163: "Discrete output point" class: instance attribute	282
Table 164: "Discrete output point" class: services	282
Table 165: "Discrete output point extended 1" class: class attribute	283
Table 166: "Discrete output point extended 1" class: instance attribute	283
Table 167: "Discrete output point extended 1" class: services	283
Table 168: "Discrete output point extended 2" class: class attribute	284
Table 169: "Discrete output point extended 2" class: instance attribute	284
Table 170: "Discrete output point extended 2" class: services	284
Table 171: "Discrete output point extended 3" class: class attribute	285
Table 172: "Discrete output point extended 3" class: instance attribute	285
Table 173: "Discrete output point extended 3" class: services	285
Table 174: "Analog input point" class: class attribute	286
Table 175: "Analog input point" class: instance attribute	286
Table 176: "Analog input point" class: services	286
Table 177: "Analog input point extended 1" class: class attribute	287
Table 178: "Analog input point extended 1" class: instance attribute	287
Table 179: "Analog input point extended 1" class: services	287
Table 180: "Analog input point extended 2" class: class attribute	288
Table 181: "Analog input point extended 2" class: instance attribute	288
Table 182: "Analog input point extended 2" class: services	288
Table 183: "Analog input point extended 3" class: class attribute	289
Table 184: "Analog input point extended 3" class: instance attribute	289
Table 185: "Analog input point extended 3" class: services	289
Table 186: "Analog output point" class: class attribute	290
Table 187: "Analog output point" class: instance attribute	290
Table 188: "Analog output point" class: services	290
Table 189: "Analog output point extended 1" class: class attribute	291
Table 190: "Analog output point extended 1" class: instance attribute	291
Table 191: "Analog output point extended 1" class: services	291
Table 192: "Analog output point extended 2" class: class attribute	292
Table 193: "Analog output point extended 2" class: instance attribute	292

Table 194: "Analog output point extended 2" class: services	292
Table 195: "Analog output point extended 3" class: class attribute	293
Table 196: "Analog output point extended 3" class: instance attribute	293
Table 197: "Analog output point extended 3" class: services	293
Table 204: "IOM Input Point Class" class: class attributes	294
Table 206: "IOM Input Point Class" class: services	294
Table 207: "IOM Output Point Class" class: class attributes	295
Table 209: "IOM Output Point Class" class: services	295
Table 198: "Node information" class: class attribute	296
Table 199: "Node information" class: instance attribute	296
Table 200: "Node information" class: services	296
Table 201: "Input Fieldbus Variable USINT" class: class attribute	297
Table 202: "Input Fieldbus Variable USINT" class: instance attribute	297
Table 203: "Input Fieldbus Variable USINT" class: services	297
Table 204: "Input Fieldbus Variable USINT Extended 1" class: class attribute	298
Table 205: "Input Fieldbus Variable USINT Extended 1" class: instance attribute	298
Table 206: "Input Fieldbus Variable USINT Extended 1" class: services	298
Table 207: "Input Fieldbus Variable USINT Extended 2" class: class attribute	299
Table 208: "Input Fieldbus Variable USINT Extended 2" class: instance attribute	299
Table 209: "Input Fieldbus Variable USINT Extended 2" class: services	299
Table 210: "Input Fieldbus Variable UINT" class: class attribute	300
Table 211: "Input Fieldbus Variable UINT" class: instance attribute	300
Table 212: "Input Fieldbus Variable UINT" class: services	300
Table 213: "Input Fieldbus Variable UINT Extended 1" class: class attribute	301
Table 214: "Input Fieldbus Variable UINT Extended 1" class: instance attribute	301
Table 215: "Input Fieldbus Variable UINT Extended 1" class: services	301
Table 216: "Input Fieldbus Variable UDINT" class: class attribute	302
Table 217: "Input Fieldbus Variable UDINT" class: instance attribute	302
Table 218: "Input Fieldbus Variable UDINT" class: services	302
Table 219: "Input Fieldbus Variable UDINT Offset UINT" class: class attribute	303
Table 220: "Input Fieldbus Variable UDINT Offset UINT" class: instance attribute	303
Table 221: "Input Fieldbus Variable UDINT Offset UINT" class: services	303
Table 222: "Output Fieldbus Variable USINT" class: class attribute	304
Table 223: "Output Fieldbus Variable USINT" class: instance attribute	304
Table 224: "Output Fieldbus Variable USINT" class: services	304
Table 225: "Output Fieldbus Variable USINT Extended 1" class: class attribute	305
Table 226: "Output Fieldbus Variable USINT Extended 1" class: instance attribute	305
Table 227: "Output Fieldbus Variable USINT Extended 1" class: services	305
Table 228: "Fieldbus Output Variable Byte 2 Class (A5 _{hex})" class: class attribute	306
Table 229: "Fieldbus Output Variable Byte 2 Class (A5 _{hex})" class: instance attribute	306
Table 230: "Fieldbus Output Variable Byte 2 Class (A5 _{hex})" class: services	306

Table 231: "Fieldbus Output Variable Word Class (A8 _{hex})" class:	
class attribute.....	307
Table 232: "Fieldbus Output Variable Word Class (A8 _{hex})" class:	
instance attribute	307
Table 233: "Fieldbus Output Variable Word Class (A8 _{hex})" class: services.....	307
Table 234: "Fieldbus Output Variable Word 1 Class (A9 _{hex})" class:	
class attribute.....	308
Table 235: "Fieldbus Output Variable Word 1 Class (A9 _{hex})" class:	
instance attribute	308
Table 236: "Fieldbus Output Variable Word 1 Class (A9 _{hex})" class: services...	308
Table 237: "Fieldbus Output Variable Double-Word Class (AC _{hex})" class:	
class attribute.....	309
Table 238: "Fieldbus Output Variable Double-Word Class (AC _{hex})" class:	
instance attribute	309
Table 239: "Fieldbus Output Variable Double-Word Class (AC _{hex})" class:	
services.....	309
Table 240: "Fieldbus Output Variable Double-Word 1 Class (AD _{hex})" class:	
class attribute.....	310
Table 241: "Fieldbus Output Variable Double-Word 1 Class (AD _{hex})" class:	
instance attribute	310
Table 242: "Fieldbus Output Variable Double-Word 1 Class (AD _{hex})" class:	
services	310
Table 243: Libraries	311
Table 244: S-BUS cable, assembled on both ends, B coded.....	312
Table 245: S-BUS terminator (B coded) and USB cable.....	312
Table 246: Supply cable, assembled on both ends, A coded	313
Table 247: Ethernet cable, assembled on one end, D coded	313
Table 248: Ethernet cable, assembled on both ends, D coded	314
Table 249: Ethernet accessories	314
Table 250: Mounting rail adapters and profile adapters.....	314
Table 251: Protective caps for unassigned connections sockets	315

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 D-32385 Minden
Hansastraße 27
D-32423 Minden
Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 5 71/8 87 – 1 69
E-mail: info@wago.com
Web: <http://www.wago.com>

