

WAGO-I/O-SYSTEM 750

Manual



750-831

BACnet/IP Controller

Version 1.2.1

© 2017 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

The contents of this documentation are taken in part from the BACnet Standard 135-2010 or are based on the original contents. These contents are subject to copyright.

The following applies to these contents: ©2010, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org).
Reprinted by permission from 2010 ASHRAE Standard-135. This material may not be copied nor distributed in either paper or digital form without ASHRAE's permission.

The following applies for the BACnet logo: BACnet® is a registered trademark of ASHRAE.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 7 77
Fax: +49 (0) 571/8 87 – 87 77

E-Mail: tcba@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

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

Table of Contents

1	Notes about this Documentation.....	13
1.1	Validity of this Documentation.....	13
1.2	Copyright.....	13
1.3	Symbols.....	14
1.4	Number Notation.....	16
1.5	Font Conventions	16
2	Important Notes	17
2.1	Legal Bases	17
2.1.1	Subject to Changes	17
2.1.2	Personnel Qualifications.....	17
2.1.3	Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions.....	17
2.1.4	Technical Condition of Specified Devices	18
2.2	Safety Advice (Precautions).....	19
2.3	Special Use Conditions for ETHERNET Devices	21
3	System Description.....	22
3.1	Manufacturing Number	23
3.2	Hardware Address (MAC ID).....	24
3.3	Component Update.....	25
3.4	Storage, Assembly and Transport	25
3.5	Assembly Guidelines/Standards.....	25
3.6	Power Supply	26
3.6.1	Fuse Protection of the Electronic Circuit Power Supply.....	26
3.6.2	Isolation	27
3.6.3	System Supply	28
3.6.3.1	Connection.....	28
3.6.3.2	Dimensioning.....	29
3.6.4	Field Supply.....	32
3.6.4.1	Connection.....	32
3.6.4.2	Fusing	34
3.6.5	Supplementary Power Supply Regulations	37
3.6.6	Supply Example.....	38
3.6.7	Power Supply Unit	40
3.7	Grounding	41
3.7.1	Grounding the DIN Rail	41
3.7.1.1	Framework Assembly	41
3.7.1.2	Insulated Assembly.....	41
3.7.2	Grounding Function.....	42
3.8	Shielding	43
3.8.1	General	43
3.8.2	Bus Cables	43
3.8.3	Shielded Signal Lines	44
3.8.4	WAGO Shield Connecting System	44

4	Device Description	45
4.1	View	49
4.2	Connectors.....	51
4.2.1	Device Supply	51
4.2.2	Fieldbus Connection.....	52
4.3	Display Elements	53
4.4	Operating Elements.....	55
4.4.1	Service Interface.....	55
4.4.2	Mode Selector Switch.....	56
4.4.3	Address Selection Switch.....	58
4.4.4	Memory Card Slot	59
4.4.4.1	Inserting a Memory Card.....	60
4.4.4.2	Removing the Memory Card	60
4.5	Technical Data	61
4.5.1	Device Data	61
4.5.2	System Data.....	61
4.5.3	Supply.....	62
4.5.4	Fieldbus MODBUS/TCP.....	62
4.5.5	Accessories.....	63
4.5.6	Connection Type	63
4.5.7	Climatic Environmental Conditions.....	63
4.5.8	Mechanical Strength.....	64
4.6	Approvals	65
4.7	Standards and Guidelines.....	66
4.8	BACnet-Building-Controller (B-BC).....	67
4.8.1	Data Sharing BIBBs	69
4.8.2	Alarm and Event Management BIBBs.....	71
4.8.3	Scheduling BIBBs	72
4.8.4	Trending BIBBs.....	73
4.8.5	Device- und Network-Management-BIBBs.....	74
4.8.6	“Native” Operation of the B-BC	78
4.8.6.1	Analog Input Object	79
4.8.6.2	Analog Output Object.....	80
4.8.6.3	Binary Input Object	81
4.8.6.4	Binary Output Object.....	82
4.8.6.5	Calendar Object	83
4.8.6.6	Device Object	84
4.8.6.7	File Object	85
4.8.6.8	Schedule Object	85
5	Mounting.....	87
5.1	Installation Position.....	87
5.2	Overall Configuration	87
5.3	Mounting onto Carrier Rail.....	89
5.3.1	Carrier Rail Properties.....	89
5.3.2	WAGO DIN Rails	90
5.4	Spacing.....	90
5.5	Mounting Sequence.....	91
5.6	Inserting and Removing Devices	92
5.6.1	Inserting the Fieldbus Coupler/Controller.....	93

5.6.2	Removing the Fieldbus Coupler/Controller	93
5.6.3	Inserting the I/O Module	94
5.6.4	Removing the I/O Module.....	95
6	Connect Devices	96
6.1	Data Contacts/Internal Bus	96
6.2	Power Contacts/Field Supply	97
6.3	Connecting a Conductor to the CAGE CLAMP®	98
7	Function Description	99
7.1	Operating System	99
7.1.1	Run-up	99
7.1.2	PFC Cycle.....	99
7.2	Process Data Architecture	101
7.2.1	Basic Structure.....	101
7.2.2	Example of an Input Process Image	103
7.2.3	Example of an Output Data Process Image.....	104
7.2.4	MODBUS Process Data	105
7.3	Data Exchange	106
7.3.1	MODBUS Memory Areas.....	107
7.3.2	Addressing.....	110
7.3.2.1	Addressing of I/O Modules	110
7.3.2.2	Example of Addressing.....	111
7.3.2.3	IEC-61131-3 Address Areas.....	113
7.3.2.4	Absolute Addressing.....	113
7.3.3	Data Exchange between MODBUS/TCP Master and I/O Modules.	115
7.3.4	Data Exchange between PLC Function (CPU) and I/O Modules ...	116
7.3.5	Data Exchange between Master and PLC Function (CPU).....	117
7.3.5.1	Example of MODBUS/TCP Master and PLC Function (CPU) ..	117
7.3.6	Application Example	119
7.4	Memory Card Function	120
7.4.1	Project Copying and Automatic System Start.....	122
7.4.2	Back-up Function (Storing Device-internal Data and Settings on the Memory Card)	124
7.4.3	Restore Function (Loading Device-internal Data and Settings from the Memory Card)	125
7.4.4	Saving BACnet Trend Data.....	128
7.4.5	Inserting a Memory Card During Operation	129
7.4.6	Removing the Memory Card During Operation.....	131
7.4.7	Saving WAGO-I/O-PRO project to Memory Card	132
7.4.8	FTP Network Access to the File System of the Memory Card	135
7.4.9	Directory Structure	136
8	Commissioning	137
8.1	Connecting Client PC and Fieldbus Nodes	138
8.2	Allocating the IP Address to the Fieldbus Node	138
8.2.1	Information about the Starting Behavior regarding BACnet and WAGO-I/O-PRO Program	140
8.2.2	Assigning IP Address via Address Selection Switch	141
8.2.3	Assigning IP Address via WAGO-ETHERNET-Settings.....	143
8.2.4	Assigning the IP Address with a PLC program.....	145

8.2.5	Assigning the IP Address with a BootP Server	145
8.2.5.1	Note MAC ID	146
8.2.5.2	Determining IP addresses	148
8.2.5.3	Assigning the IP address.....	149
8.2.5.4	Assigning the IP address permanently by option “use IP from EEPROM“	149
8.2.5.5	Reasons for Failed IP Address Assignment.....	153
8.3	Testing the Function of the Fieldbus Node	154
8.4	Preparing the Flash File System.....	156
8.5	Synchronizing the Real-Time Clock.....	158
8.6	Restoring Factory Settings	161
9	Programming the PFC Using WAGO-I/O-PRO	162
9.1	Configuring the Fieldbus Controller using the I/O Configurator.....	165
9.1.1	Configuration using the “EA-config.xml” File	168
9.2	Automatically prepare SYM_XML file	170
9.3	Create SYM_XML file manually.....	173
9.4	ETHERNET Libraries for WAGO-I/O-PRO.....	174
9.5	Functional Restrictions and Limits	176
9.6	General Information about IEC Tasks	178
9.6.1	IEC Task Sequence.....	180
9.6.2	Overview of Most Important Task Priorities.....	180
9.7	System Events	182
9.7.1	Enabling/Disabling System Events	182
9.8	Transfer the IEC program to the Fieldbus controller	184
9.8.1	Transfer via Serial Service Port.....	186
9.8.2	Transfer via ETHERNET	189
10	Configuring via the Web-Based Management System (WBM).....	191
10.1	Information.....	193
10.2	Ethernet	195
10.3	TCP/IP.....	198
10.4	Port	200
10.5	SNMP	202
10.5.1	SNMP V1/V2c.....	203
10.5.2	SNMP V3	205
10.6	Watchdog	207
10.7	Clock	209
10.8	Security	212
10.9	MODBUS.....	215
10.10	BACnet.....	217
10.11	PLC Info.....	219
10.12	PLC	220
10.13	Features	225
10.14	I/O Config	227
10.15	Disk Info.....	230
10.16	SD Card	231
10.17	Backup & Restore	233
10.18	WebVisu.....	236
11	Diagnostics	238

11.1	LED Signaling.....	238
11.1.1	Evaluating Fieldbus Status	239
11.1.2	Evaluating Node Status – I/O LED (Blink Code Table)	240
11.1.2.1	USR LED.....	251
11.1.3	Evaluating Memory Card Status	251
11.1.4	Evaluating Power Supply Status	251
11.2	Fault Behavior	252
11.2.1	Loss of Fieldbus	252
11.2.2	Internal Data Bus Failure.....	253
12	Fieldbus Communication	254
12.1	Implemented Protocols.....	254
12.1.1	Communication Protocols	254
12.1.1.1	IP (Internet Protocol)	254
12.1.1.2	TCP (Transmission Control Protocol)	259
12.1.1.3	UDP (User Datagram Protocol).....	259
12.1.2	Configuration and Diagnostics Protocols.....	260
12.1.2.1	BootP (Bootstrap Protocol).....	260
12.1.2.2	DHCP (Dynamic Host Configuration Protocol).....	261
12.1.2.3	HTTP (Hypertext Transfer Protocol).....	264
12.1.2.4	DNS (Domain Name Systems).....	264
12.1.2.5	SNTP-Client (Simple Network Time Protocol)	264
12.1.2.6	FTP-Server (File Transfer Protocol).....	265
12.1.2.7	SNMP (Simple Network Management Protocol).....	266
12.1.2.7.1	MIB II Description.....	268
12.1.2.7.2	Traps.....	268
12.1.3	Application Protocols	269
12.1.3.1	BACnet/IP.....	269
12.1.3.1.1	Interoperability.....	269
12.1.3.1.2	BACnet Components	270
12.1.3.1.2.1	Objects.....	270
12.1.3.1.2.2	Properties.....	271
12.1.3.1.2.3	Services	271
12.1.3.1.2.3.1	Client-Server Communication.....	276
12.1.3.1.2.3.2	Prioritization	276
12.1.3.1.2.4	Interoperability Area (IA)	280
12.1.3.1.2.5	BACnet Interoperability Building Blocks - BIBBs.....	282
12.1.3.1.2.6	Protocol Implementation Conformance Statement - PICS	283
12.1.3.1.2.7	Device Profile.....	284
12.1.3.1.2.8	Data Types.....	285
12.1.3.1.2.9	BACnet Components in Overview.....	286
12.1.3.1.3	Scope of Use in Building Automation	287
12.1.3.1.3.1	Management Level.....	287
12.1.3.1.3.2	Automation Level.....	288
12.1.3.1.3.3	Field Level.....	288
12.1.3.1.4	BACnet in the ISO/OSI Model	288
12.1.3.1.5	BACnet in the Network	290
12.1.3.1.5.1	IP Message Tunneling.....	290
12.1.3.1.5.2	BACnet/IP	291
12.1.3.1.5.3	BACnet/IP and Unicast/Broadcast	291
12.1.3.1.5.4	BACnet/IP in Foreign Networks	292

12.2	MODBUS Functions	293
12.2.1	General	293
12.2.2	Use of the MODBUS Functions	296
12.2.3	Description of the MODBUS Functions	297
12.2.3.1	Function Code FC1 (Read Coils)	298
12.2.3.2	Function Code FC2 (Read Discrete Inputs).....	300
12.2.3.3	Function Code FC3 (Read Multiple Registers)	302
12.2.3.4	Function Code FC4 (Read Input Registers).....	303
12.2.3.5	Function Code FC5 (Write Coil)	304
12.2.3.6	Function Code FC6 (Write Single Register)	305
12.2.3.7	Function Code FC11 (Get Comm Event Counter)	306
12.2.3.8	Function Code FC15 (Write Multiple Coils).....	307
12.2.3.9	Function Code FC16 (Write Multiple Registers)	309
12.2.3.10	Function Code FC22 (Mask Write Register).....	310
12.2.3.11	Function Code FC23 (Read/Write Multiple Registers)	311
12.2.4	MODBUS Register Mapping	313
12.2.5	MODBUS Registers	316
12.2.5.1	Accessing Register Values	317
12.2.5.2	Watchdog Registers	317
12.2.5.3	Diagnostic Registers	323
12.2.5.4	Configuration Registers	324
12.2.5.5	Firmware Information Registers.....	329
12.2.5.6	Constant Registers	331
13	I/O Modules	333
13.1	Overview	333
13.2	Process Data Architecture for MODBUS/TCP	334
13.2.1	Digital Input Modules.....	335
13.2.1.1	1 Channel Digital Input Module with Diagnostics	335
13.2.1.2	2 Channel Digital Input Modules	335
13.2.1.3	2 Channel Digital Input Module with Diagnostics	335
13.2.1.4	2 Channel Digital Input Module with Diagnostics and Output Process Data.....	336
13.2.1.5	4 Channel Digital Input Modules	336
13.2.1.6	8 Channel Digital Input Modules	336
13.2.1.7	8 Channel Digital Input Module PTC with Diagnostics and Output Process Data.....	337
13.2.1.8	16 Channel Digital Input Modules	337
13.2.2	Digital Output Modules.....	338
13.2.2.1	1 Channel Digital Output Module with Input Process Data	338
13.2.2.2	2 Channel Digital Output Modules.....	338
13.2.2.3	2 Channel Digital Input Modules with Diagnostics and Input Process Data.....	339
13.2.2.4	4 Channel Digital Output Modules.....	340
13.2.2.5	4 Channel Digital Output Modules with Diagnostics and Input Process Data.....	340
13.2.2.6	8 Channel Digital Output Module	340
13.2.2.7	8 Channel Digital Output Modules with Diagnostics and Input Process Data.....	341
13.2.2.8	16 Channel Digital Output Modules.....	341
13.2.2.9	8 Channel Digital Input/Output Modules	342

13.2.3	Analog Input Modules.....	343
13.2.3.1	1 Channel Analog Input Modules.....	343
13.2.3.2	2 Channel Analog Input Modules.....	343
13.2.3.3	4 Channel Analog Input Modules.....	344
13.2.3.4	3-Phase Power Measurement Module	345
13.2.3.5	8 Channel Analog Input Modules.....	346
13.2.4	Analog Output Modules	347
13.2.4.1	2 Channel Analog Output Modules	347
13.2.4.2	4 Channel Analog Output Modules	347
13.2.4.3	8 Channel Analog Output Modules	348
13.2.5	Specialty Modules	349
13.2.5.1	Counter Modules	349
13.2.5.2	Pulse Width Modules.....	351
13.2.5.3	Serial Interface Modules with alternative Data Format.....	351
13.2.5.4	Serial Interface Modules with Standard Data Format	352
13.2.5.5	Data Exchange Module.....	352
13.2.5.6	SSI Transmitter Interface Modules.....	352
13.2.5.7	Incremental Encoder Interface Modules.....	353
13.2.5.8	DC-Drive Controller	355
13.2.5.9	Stepper Controller.....	356
13.2.5.10	RTC Module	357
13.2.5.11	DALI/DSI Master Module.....	357
13.2.5.12	DALI Multi-Master Module	358
13.2.5.13	LON [®] FTT Module.....	360
13.2.5.14	EnOcean Radio Receiver	360
13.2.5.15	MP Bus Master Module.....	360
13.2.5.16	Bluetooth [®] RF-Transceiver.....	361
13.2.5.17	Vibration Velocity/Bearing Condition Monitoring VIB I/O	362
13.2.5.18	KNX/EIB/TP1 Module	362
13.2.5.19	AS-interface Master Module	363
13.2.6	System Modules	365
13.2.6.1	System Modules with Diagnostics.....	365
13.2.6.2	Binary Space Module	365
14	Application Examples.....	366
14.1	Test of MODBUS protocol and fieldbus nodes	366
14.2	Visualization and Control using SCADA Software.....	366
15	Use in Hazardous Environments	369
15.1	Marking Configuration Examples.....	370
15.1.1	Marking for Europe According to ATEX and IEC-Ex	370
15.1.2	Marking for America According to NEC 500.....	375
15.2	Installation Regulations.....	376
15.2.1	Special Conditions for Safe Use (ATEX Certificate TÜV 07 ATEX 554086 X).....	377
15.2.2	Special Conditions for Safe Use (ATEX Certificate TÜV 12 ATEX 106032 X).....	378
15.2.3	Special Conditions for Safe Use (IEC-Ex Certificate TUN 09.0001 X).....	379
15.2.4	Special Conditions for Safe Use (IEC-Ex Certificate IECEX TUN 12.0039 X).....	380
15.2.5	Special Conditions for Safe Use According to ANSI/ISA 12.12.01	381

16	Appendix.....	382
16.1	List of all BACnet Properties in Native Operation	382
16.1.1	Acked_Transitions.....	383
16.1.2	Active_COV_Subscriptions	384
16.1.3	Active_Text	384
16.1.4	Alarm_Value	385
16.1.5	APDU_Segment_Timeout.....	385
16.1.6	APDU_Timeout.....	386
16.1.7	Application_Software_Version	386
16.1.8	Archive	387
16.1.9	Backup_Failure_Timeout	387
16.1.10	Change_Of_State_Count.....	387
16.1.11	Change_Of_State_Time	388
16.1.12	Configuration_Files.....	389
16.1.13	COV_Increment	389
16.1.14	Database_Revision	389
16.1.15	Data_List	389
16.1.16	Daylight_Savings_Status.....	390
16.1.17	Deadband.....	390
16.1.18	Description	391
16.1.19	Device_Address_Binding.....	391
16.1.20	Device_Type.....	392
16.1.21	Effective_Period.....	392
16.1.22	Elapsed_Active_Time	392
16.1.23	Event_Enable.....	393
16.1.24	Event_State.....	393
16.1.25	Event_Time_Stamps.....	393
16.1.26	Exception_Schedule	394
16.1.27	Feedback_Value	395
16.1.28	File_Access_Method	395
16.1.29	File_Size	396
16.1.30	File_Type.....	396
16.1.31	Firmware_Revision	396
16.1.32	High_Limit	396
16.1.33	Inactive_Text.....	397
16.1.34	Last_Restore_Time.....	397
16.1.35	Limit_Enable	398
16.1.36	List_Of_Object_Property_References	398
16.1.37	Local_Date	398
16.1.38	Local_Time.....	399
16.1.39	Location.....	399
16.1.40	Low_Limit.....	399
16.1.41	Max_APDU_Length_Accepted	399
16.1.42	Max_Pres_Value	400
16.1.43	Max_Segments_Accepted	400
16.1.44	Message_Text (512)	400
16.1.45	Min_Pres_Value	400
16.1.46	Minimum_Off_Time	401
16.1.47	Minimum_On_Time.....	401
16.1.48	Model_Name	401

16.1.49	Modification_Date.....	402
16.1.50	Notification_Class	402
16.1.51	Notify_Type	402
16.1.52	Number_Of_APDU_Retries.....	402
16.1.53	Object_Identifier.....	403
16.1.54	Object_List	403
16.1.55	Object_Name	403
16.1.56	Object_Type	403
16.1.57	Out_Of_Service.....	404
16.1.58	Polarity	405
16.1.59	Present_Value	405
16.1.60	Priority_Array.....	408
16.1.61	Priority_For_Writing.....	408
16.1.62	Protocol_Object_Types_Supported.....	408
16.1.63	Protocol_Revision	409
16.1.64	Protocol_Services_Supported.....	409
16.1.65	Protocol_Version.....	409
16.1.66	Read_Only	410
16.1.67	Record_Count.....	410
16.1.68	Reliability	410
16.1.69	Relinquish_Default.....	411
16.1.70	Resolution.....	412
16.1.71	Schedule_Default	412
16.1.72	Segmentation_Supported.....	412
16.1.73	Status_Flags.....	413
16.1.74	System_Status.....	414
16.1.75	Time_Delay	414
16.1.76	Time_Of_Active_Time_Reset	415
16.1.77	Time_Of_State_Count_Reset.....	415
16.1.78	Units	415
16.1.79	Update_Interval	416
16.1.80	UTC_Offset	416
16.1.81	Vendor_Identifier	416
16.1.82	Vendor_Name.....	416
16.1.83	Weekly_Schedule	417
16.2	MIB II Groups.....	418
16.2.1	System Group	418
16.2.2	Interface Group.....	419
16.2.3	IP Group	421
16.2.4	IpRoute Table Group.....	422
16.2.5	ICMP Group	423
16.2.6	TCP Group.....	424
16.2.7	UDP Group.....	425
16.2.8	SNMP Group	426
16.3	WAGO MIB Groups	427
16.3.1	Company Group	427
16.3.2	Product Group	427
16.3.3	Versions Group.....	428
16.3.4	Real-Time Clock Group	429
16.3.5	Ethernet Group	430

16.3.6	Actual Error Group.....	430
16.3.7	PLC Project Group	431
16.3.8	Http Group.....	432
16.3.9	Ftp Group.....	432
16.3.10	Sntp Group.....	433
16.3.11	Snmp Group.....	433
16.3.12	Snmp Trap String Group	435
16.3.13	Snmp User Trap String Group.....	436
16.3.14	Plc Connection Group	436
16.3.15	Modbus Group.....	437
16.3.16	Process Image Group.....	437
16.3.17	Plc Data Group	439
Glossary		440
Literature List.....		458
List of Figures		459
List of Tables.....		462

1 Notes about this Documentation



Note

Always retain this documentation!

This documentation is part of the product. Therefore, retain the documentation during the entire service life of the product. Pass on the documentation to any subsequent user. In addition, ensure that any supplement to this documentation is included, if necessary.

1.1 Validity of this Documentation

This documentation is only applicable to the “BACnet/IP Controller” (750-831) and the variants listed in the table below.

Table 1: Variants

Item number/Variant	Designation
750-831	BACnet/IP Controller
750-831/000-002	BACnet/IP Controller ECO (max. 256 application-based objects*)

* The application-based objects include all objects subject to manipulation by the user:

- Native objects created automatically by connecting I/O modules
- Dynamic objects created via BACnet services
- Configured objects created via CODESYS or the BACnet Configurator

The product “BACnet/IP Controller” (750-831) shall only be installed and operated according to the instructions in this manual and the system description for the WAGO-I/O-SYSTEM 750.

NOTICE

Consider power layout of the WAGO-I/O-SYSTEM 750!

In addition to these operating instructions, you will also need the system description for the WAGO-I/O-SYSTEM 750, which can be downloaded at www.wago.com. There, you can obtain important information including information on electrical isolation, system power and supply specifications.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.3 Symbols

 DANGER**Personal Injury!**

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 DANGER**Personal Injury Caused by Electric Current!**

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

 WARNING**Personal Injury!**

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

 CAUTION**Personal Injury!**

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE**Damage to Property!**

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE**Damage to Property Caused by Electrostatic Discharge (ESD)!**

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.



Note**Important Note!**

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.



Information

Additional Information:

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.4 Number Notation

Table 2: Number Notation

Number Code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.5 Font Conventions

Table 3: Font Conventions

Font Type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Program Files\WAGO Software</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualifications

All sequences implemented on WAGO-I/O-SYSTEM 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

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

2.1.3 Use of the WAGO-I/O-SYSTEM 750 in Compliance with Underlying Provisions

Fieldbus couplers, fieldbus controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to actuators or higher-level control systems. Using programmable controllers, the signals can also be (pre-) processed.

The devices have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the devices in wet and dusty environments is prohibited.

Operating the WAGO-I/O-SYSTEM 750 devices in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section “Device Description” > “Standards and Guidelines” in the manual for the used fieldbus coupler/controller.

Appropriate housing (per 2014/34/EU) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

2.1.4 Technical Condition of Specified Devices

The devices to be supplied ex works are equipped with hardware and software configurations, which meet the individual application requirements. WAGO Kontakttechnik GmbH & Co. KG will be exempted from any liability in case of changes in hardware or software as well as to non-compliant usage of devices.

Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

2.2 Safety Advice (Precautions)

For installing and operating purposes of the relevant device to your system the following safety precautions shall be observed:



DANGER

Do not work on devices while energized!

All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.



DANGER

Install the device only in appropriate housings, cabinets or in electrical operation rooms!

The WAGO-I/O-SYSTEM 750 and its components are an open system. As such, install the system and its components exclusively in appropriate housings, cabinets or in electrical operation rooms. Allow access to such equipment and fixtures to authorized, qualified staff only by means of specific keys or tools.

NOTICE

Do not use in telecommunication circuits!

Only use devices equipped with ETHERNET or RJ-45 connectors in LANs. Never connect these devices with telecommunication networks.

NOTICE

Replace defective or damaged devices!

Replace defective or damaged device/module (e.g., in the event of deformed contacts), since the long-term functionality of device/module involved can no longer be ensured.

NOTICE

Protect the components against materials having seeping and insulating properties!

The components are not resistant to materials having seeping and insulating properties such as: aerosols, silicones and triglycerides (found in some hand creams). If you cannot exclude that such materials will appear in the component environment, then install the components in an enclosure being resistant to the above-mentioned materials. Clean tools and materials are imperative for handling devices/modules.

NOTICE**Clean only with permitted materials!**

Clean soiled contacts using oil-free compressed air or with ethyl alcohol and leather cloths.

NOTICE**Do not use any contact spray!**

Do not use any contact spray. The spray may impair contact area functionality in connection with contamination.

NOTICE**Do not reverse the polarity of connection lines!**

Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

NOTICE**Avoid electrostatic discharge!**

The devices are equipped with electronic components that may be destroyed by electrostatic discharge when touched. Please observe the safety precautions against electrostatic discharge per DIN EN 61340-5-1/-3. When handling the devices, please ensure that environmental factors (personnel, work space and packaging) are properly grounded.

2.3 Special Use Conditions for ETHERNET Devices

If not otherwise specified, ETHERNET devices are intended for use on local networks. Please note the following when using ETHERNET devices in your system:

- Do not connect control components and control networks to an open network such as the Internet or an office network. WAGO recommends putting control components and control networks behind a firewall.
- Limit physical and electronic access to all automation components to authorized personnel only.
- Change the default passwords before first use! This will reduce the risk of unauthorized access to your system.
- Regularly change the passwords used! This will reduce the risk of unauthorized access to your system.
- If remote access to control components and control networks is required, use a Virtual Private Network (VPN).
- Regularly perform threat analyses. You can check whether the measures taken meet your security requirements.
- Use “defense-in-depth” mechanisms in your system's security configuration to restrict the access to and control of individual products and networks.

3 System Description

The WAGO-I/O-SYSTEM 750 is a modular, fieldbus-independent input/output system (I/O system). The configuration described here consists of a fieldbus coupler/controller (1) and the modular I/O modules (2) for any signal shapes that form the fieldbus node together. The end module (3) completes the node and is required for correct operation of the fieldbus node.

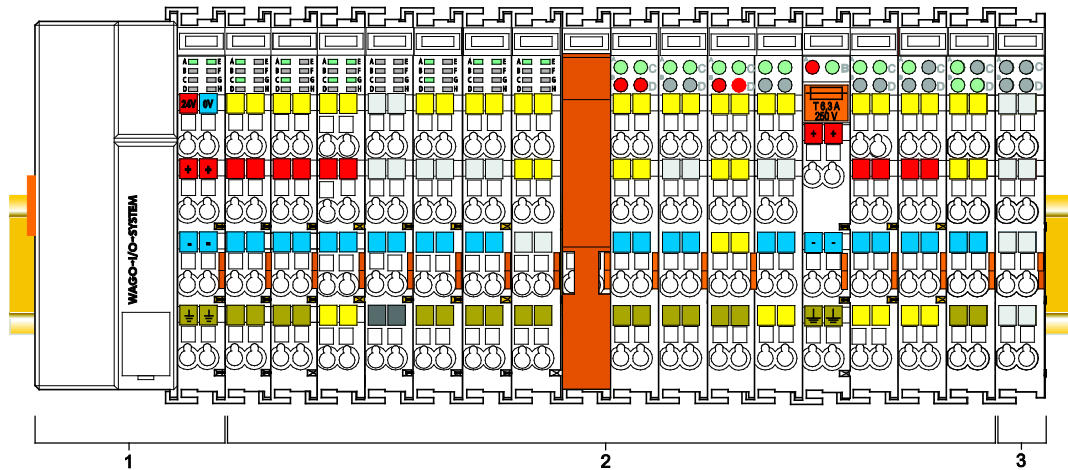


Figure 1: Fieldbus Node (Example)

Fieldbus couplers/controllers are available for different fieldbus systems.

The extended ECO fieldbus couplers contain the fieldbus interface, electronics and a power supply terminal. The fieldbus interface forms the physical interface to the relevant fieldbus. The electronics process the data of the bus modules and make it available for the fieldbus communication. The 24 V system supply and the 24 V field supply are fed in via the integrated power supply terminal.

The fieldbus coupler/controller communicates via the relevant fieldbus. The programmable fieldbus controller (PFC) enables the implementation of additional PLC functions. Programming is done with the WAGO-I/O-PRO in accordance with IEC 61131-3.

I/O modules for diverse digital and analog I/O signals as well as special functions can be connected to the fieldbus coupler/controller. The communication between the fieldbus coupler/controller and the I/O modules is carried out via an internal bus.

The components of the WAGO-I/O-SYSTEM 750 have clear termination points, light emitting diodes for status display, plug-in mini WSB tags and group marker cards for labeling.

The 1, 2 or 3 wire technology supplemented by a ground wire connection allows for direct sensor or actuator wiring.

3.1 Manufacturing Number

The serial number indicates the delivery status directly after production. This number is part of the labeling on the side of each component.

In addition, the serial number is printed on the cover cap of the configuration and programming interface of the fieldbus coupler/controller, so that it can also be read when installed.

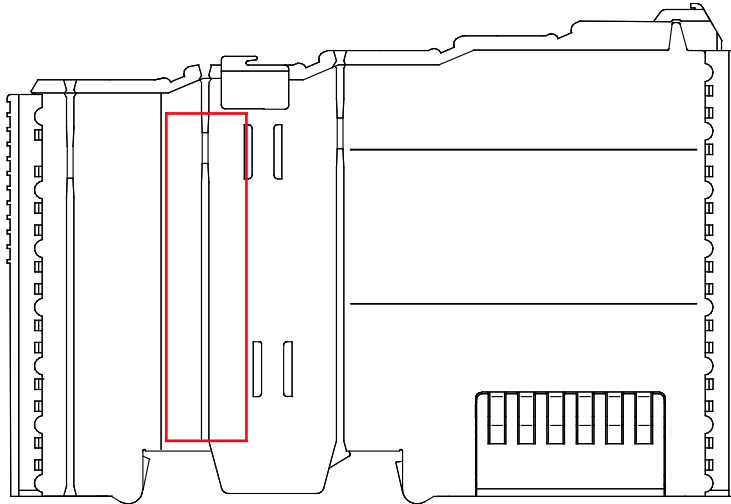


Figure 2: Marking Area for Serial Numbers

There are two serial numbers in two rows in the side marking. They are left of the release tab. The first 10 positions in the longer row of the serial numbers contain version and date identifications.

Example structure of the rows: 0114010101...

01	14	01	01	01	(additional positions)
WW	YY	FW --	HW	FL	-
Calendar week	Year	Firmware version	Hardware version	Firmware loader version	Internal information

The row order can vary depending on the production year, only the longer row is relevant. The back part of this and the shorter row contain internal administration information from the manufacturer.

3.2 Hardware Address (MAC ID)

Each BACnet/IP Controller has an internationally unambiguous physical address, referred to as the MAC-ID (Media Access Control Identity).

As part of the labeling on the right side of this component, the MAC ID is printed in the block diagram of the fieldbus coupler/controller.

In addition, the MAC ID is located on the paper strip with two self-adhesive peel-off strips on the left side of the fieldbus coupler/controller.

The MAC ID has a fixed length of 6 bytes (48 bits) which are presented hexadecimal. The first three bytes identify the manufacturer (e.g. 00:30 DE for WAGO). The second 3 bytes comprise the unique serial number of the hardware.

3.3 Component Update

For the case of an update of one component, the lateral marking on each component contains a prepared matrix.

This matrix makes columns available for altogether three updates to the entry of the current update data, like production order number (NO; starting from calendar week 13/2004), date stamp (DS), software version (SW), hardware version (HW) and the firmware loader version (FWL, if available).

Current version data for		1. Update	2. Update	3. Update	← only starting from calendar week 13/2004 ← only for fieldbus couplers/controllers
Production order no.	NO				
Date stamp	DS				
Software version	SW				
Hardware version	HW				
Firmware loader vers.	FWL				

If the update of a component took place, the current version data are registered into the columns of the matrix.

Additionally with the update of a fieldbus coupler or controller also the cover of the configuration and programming interface of the fieldbus coupler or controller is imprinted with the current production order number.

The original manufacturing information on the device's housing remains unchanged.

3.4 Storage, Assembly and Transport

Whenever possible, the components are to be stored in their original packaging. Likewise, the original packaging provides optimal protection during transport.

When assembling or repacking the components, the contacts must not be soiled or damaged. The components must be stored and transported in appropriate containers/packaging. Thereby, the ESD information is to be regarded.

3.5 Assembly Guidelines/Standards

- DIN 60204 Electrical equipment of machines
- DIN EN 50178 Electronic equipment for use in power installations (replacement for VDE 0160)
- EN 60439 Low-voltage switchgear and controlgear assemblies

3.6 Power Supply

3.6.1 Fuse Protection of the Electronic Circuit Power Supply

NOTICE

Only implement the electronic circuit power supply with a suitable fuse!

The electronic power supply of the fieldbus coupler/controller must only be connected via a slow blow 2A fuse, as shown in the following figure. The electronics may be damaged with higher currents.

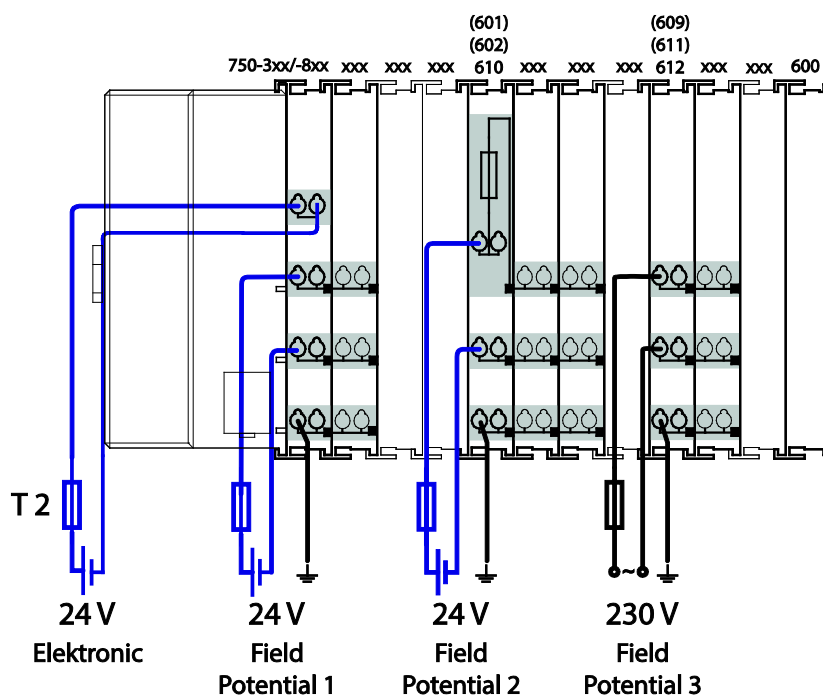


Figure 3: Fuse Protection of the Electronic Circuit Power Supply

3.6.2 Isolation

Within the fieldbus node, there are three electrically isolated potentials:

- Electrically isolated fieldbus interface via transformer
- Electronics of the fieldbus couplers/controllers and the I/O modules (internal bus)
- All I/O modules have an electrical isolation between the electronics (internal bus, logic) and the field electronics. Some digital and analog input modules have each channel electrically isolated, please see catalog.

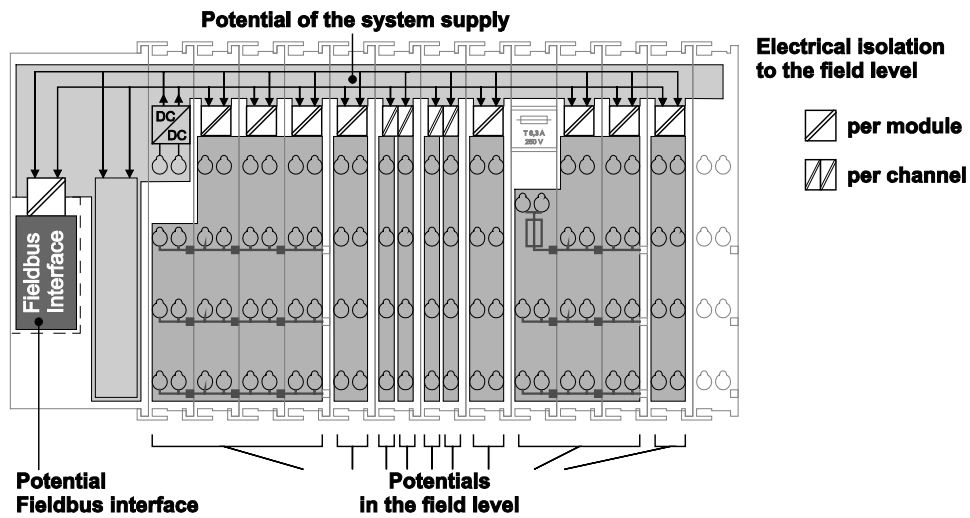


Figure 4: Isolation for Fieldbus Couplers/Controllers (Example)

3.6.3 System Supply

3.6.3.1 Connection

The WAGO-I/O-SYSTEM 750 requires a 24 V direct current system supply. The power supply is provided via the fieldbus coupler/controller and, if necessary, in addition via internal system supply modules 750-613. The power supply is reverse voltage protected.

NOTICE

Do not use an incorrect voltage/frequency!

The use of an incorrect supply voltage or frequency can cause severe damage to the components.

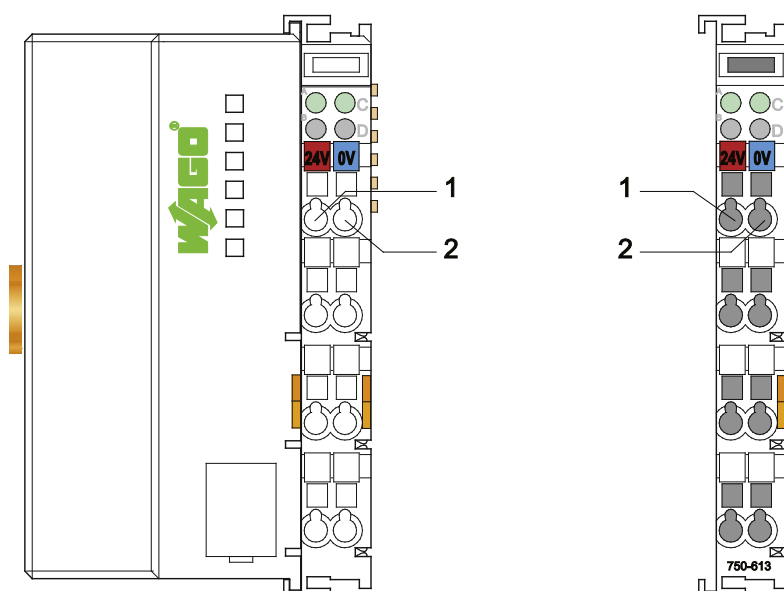


Figure 5: System Supply via Fieldbus Coupler/Controller (left) and via Internal System Supply Module (right)

Table 4: Legend for Figure “System Supply via Fieldbus Coupler/Controller (left) and via Internal System Supply Module (right)”

Position	Description
1	System supply DC 24 V (-25 % ... +30 %)
2	System supply 0 V

The fed DC 24 V supplies all internal system components, e.g. fieldbus coupler/controller electronics, fieldbus interface and I/O modules via the internal bus (5 V system voltage). The 5 V system voltage is galvanically connected to the 24 V system supply.

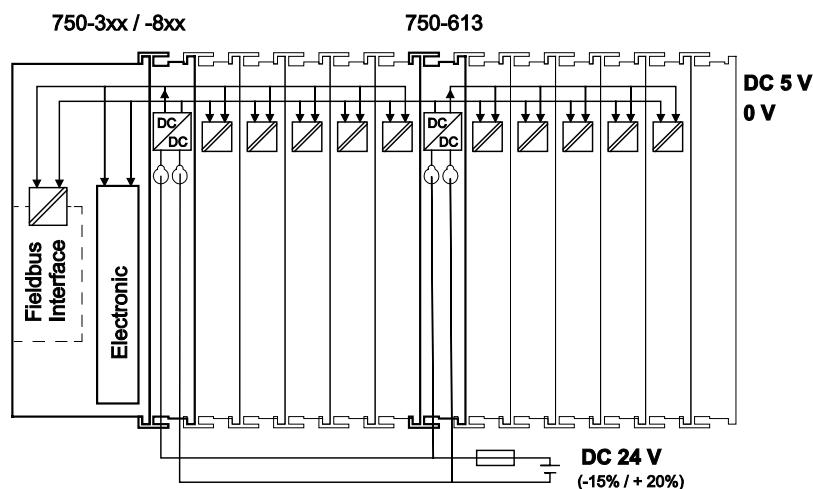


Figure 6: System Voltage for Standard Couplers/Controllers and Extended ECO Couplers

Note



Only reset the system simultaneously for all supply modules!

Reset the system by switching the system supply simultaneously at all supply modules (fieldbus coupler/controller and potential supply module with bus power supply) off and on again.

3.6.3.2 Dimensioning

Note



Recommendation

A stable power supply cannot always be assumed. Therefore, you should use regulated power supplies to ensure the quality of the supply voltage.

The supply capacity of the fieldbus coupler/controller or the internal system supply module can be taken from the technical data of the components.

Table 5: Alignment

Internal current consumption^{*)}	Current consumption via system voltage (5 V for electronics of I/O modules and fieldbus coupler/controller).
Total current for I/O modules^{*)}	Available current for the I/O modules. Provided by the bus power supply unit. See fieldbus coupler/controller and internal system supply module

^{*)} See current catalog, manuals, Internet

Example:**Calculating the current consumption on an Example Coupler:**

Internal current consumption	380 mA at 5 V
<u>Residual current for bus modules</u>	<u>1620 mA at 5 V</u>
Sum $I_{(5\text{ V})}$ total	2000 mA at 5V

The internal current consumption is indicated in the technical data for each bus terminal. In order to determine the total requirement, add together the values of all I/O modules in the node.

Note

Please note the aggregate current for I/O modules. It may be necessary to supply potential!

When the sum of the internal current consumption for the I/O modules exceeds their aggregate current, you must use a supply module with bus power supply. Install it before the position where the permissible aggregate current would be exceeded.

Example:**Calculating the total current on the example coupler described above:**

A node with the example coupler, which is described above, consists of: 20 relay modules (750-517) and 10 digital input modules (750-405).

Internal current consumption	$20 \times 90 \text{ mA} = 1800 \text{ mA at } 5 \text{ V}$
	$+ 10 \times 2 \text{ mA} = 20 \text{ mA at } 5 \text{ V}$
Sum	1820 mA at 5 V

The example coupler can only provide 1620 mA (see previous example) for the I/O modules. This value is given in the associated data sheet. Consequently, an internal system supply module with bus power supply (750-613), e. g. in the middle of the node, should be added.

Note**Recommendation**

Utilize the **smartDESIGNER** feature WAGO ProServe[®] software to configure fieldbus node assembly. You can test the configuration via the integrated plausibility check.

The maximum input current of the 24 V system supply is 500 mA. The exact electrical consumption ($I_{(V)}$) can be determined with the following formulas:

Fieldbus coupler or controller

$I_{(5\text{ V})\text{ total}}$ = Sum of all the internal current consumption of the connected I/O modules + internal current consumption of the fieldbus coupler/controller

Internal system supply module

$I_{(5\text{ V})\text{ total}}$ = Sum of all the internal current consumption of the connected I/O modules at internal system supply module

$$\text{Input current } I_{(24\text{ V})} = \frac{5\text{ V}}{24\text{ V}} \times \frac{I_{(5\text{ V})\text{ total}}}{\eta}$$

$$\eta = 0.87$$

(87 % Efficiency of the power supply at nominal load 24 V)



Note

Activate all outputs when testing the current consumption!

If the electrical consumption of a power supply point for the 24 V system supply exceeds 500 mA, then the cause may be an improperly dimensioned node or a defect.

During the test, you must activate all outputs.

3.6.4 Field Supply

3.6.4.1 Connection

Sensors and actuators can be directly connected to the relevant channel of the I/O module in 1, 2, 3 or 4 conductor connection technology. The I/O module supplies power to the sensors and actuators. The input and output drivers of some I/O modules require the field side supply voltage.

The fieldbus coupler/controller provides field side power (DC 24 V). In this case it is a passive power supply without protection equipment.

Power supply modules with or without fuse holder and diagnostic capability are available for the power supply of other field potentials (DC 24 V, AC/DC 0 ... 230 V, AC 120 V, AC 230 V). The power supply modules can also be used to set up various potential groups. The connections are connected in pairs to a power contact.

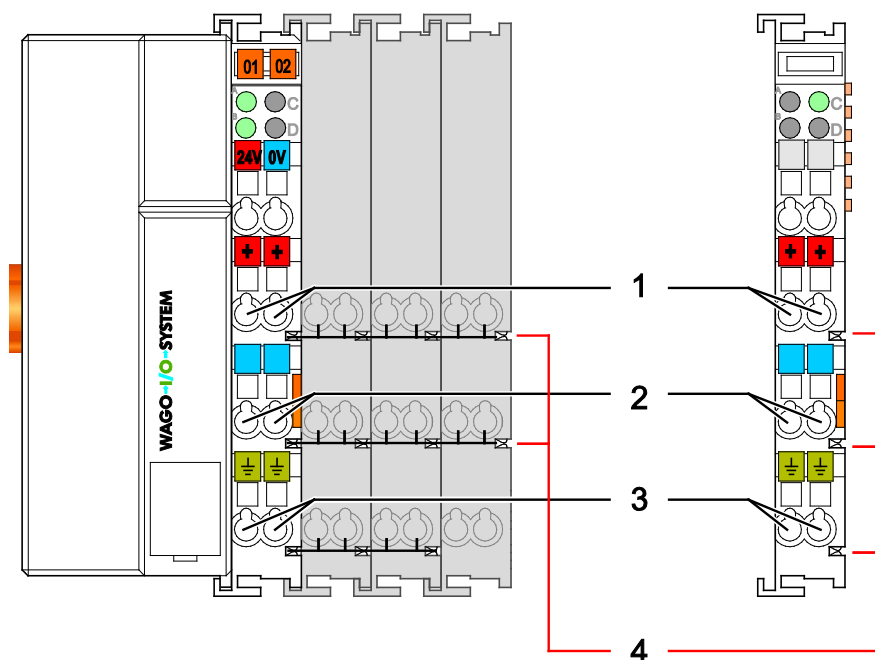


Figure 7: Field Supply for Standard Couplers/Controllers and Extended ECO Couplers

Table 6: Legend for Figure “Field Supply for Standard Couplers/Controllers and Extended ECO Couplers”

Field supply	
1	24 V (-15 % / +20 %)
2	0 V
3	Optional ground potential
Power jumper contacts	
4	Potential distribution to adjacent I/O modules

The field-side power supply is automatically derived from the power jumper contacts when snapping an I/O module.

The current load of the power contacts must not exceed 10 A on a continual basis.

By inserting an additional power supply module, the field supply via the power contacts is disrupted. From there a new power supply occurs which may also contain a new voltage potential.

Note



Re-establish the ground connection when the connection to the power jumper contacts is disrupted!

Some I/O modules have no or very few power contacts (depending on the I/O function). Due to this, the passing through of the relevant potential is disrupted. If you require a field supply via power jumper contacts for subsequent I/O modules, then you have to use a power supply module.

Note the data sheets of the I/O modules.

Note



Use a spacer module when setting up a node with different potentials!

In the case of a node setup with different potentials, e.g. the alteration from DC 24 V to AC 230 V, you should use a spacer module. The optical separation of the potentials acts as a warning to heed caution in the case of wiring and maintenance works. Thus, you can prevent the results of wiring errors.

3.6.4.2 Fusing

Internal fusing of the field supply is possible for various field voltages via an appropriate power supply module.

Table 7: Power Supply Modules

Order No.	Field Voltage
750-601	24 V DC, Supply/Fuse
750-609	230 V AC, Supply/Fuse
750-615	120 V AC, Supply/Fuse
750-617	24 V AC, Supply/Fuse
750-610	24 V DC, Supply/Fuse/Diagnosis
750-611	230 V AC, Supply/Fuse/Diagnosis
750-606	Supply Module 24 V DC, 1,0 A, Ex i
750-625/000-001	Supply Module 24 V DC, 1,0 A, Ex i (without diagnostics)

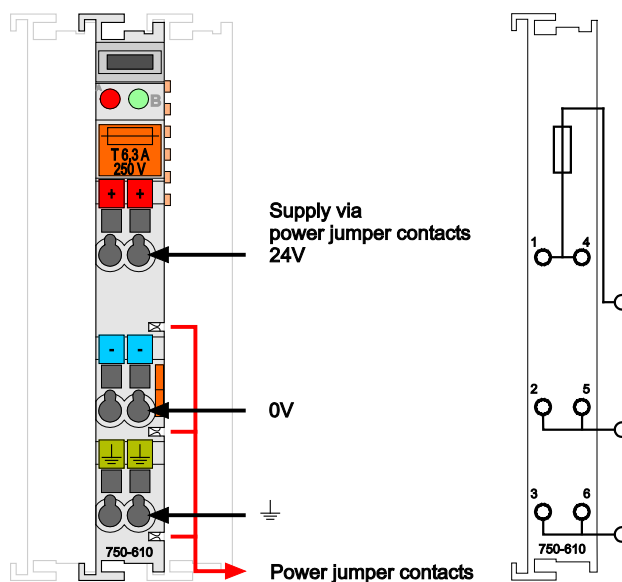


Figure 8: Supply Module with Fuse Carrier (Example 750-610)

NOTICE

Observe the maximum power dissipation and, if required, UL requirements!

In the case of power supply modules with fuse holders, you must only use fuses with a maximum dissipation of 1.6 W (IEC 127).

For UL approved systems only use UL approved fuses.

In order to insert or change a fuse, or to switch off the voltage in succeeding I/O modules, the fuse holder may be pulled out. In order to do this, use a screwdriver for example, to reach into one of the slits (one on both sides) and pull out the holder.



Figure 9: Removing the Fuse Carrier

Lifting the cover to the side opens the fuse carrier.

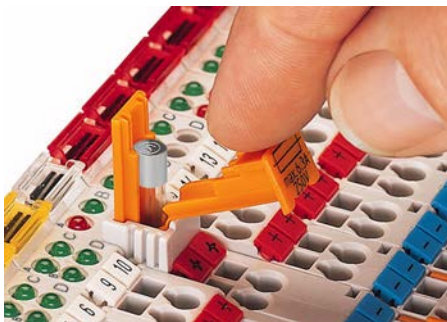


Figure 10: Opening the Fuse Carrier



Figure 11: Changing the Fuse

After changing the fuse, the fuse carrier is pushed back into its original position.

Alternatively, fusing can be done externally. The fuse modules of the WAGO series 281 and 282 are suitable for this purpose.

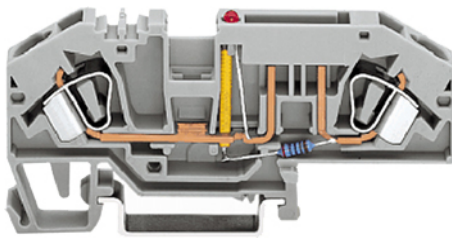


Figure 12: Fuse Modules for Automotive Fuses, Series 282

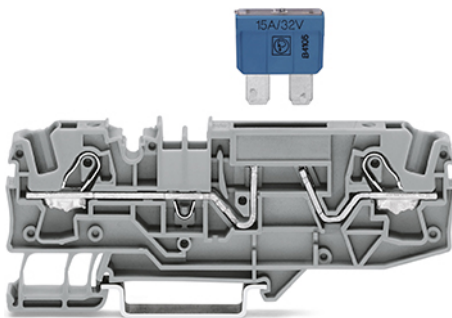


Figure 13: Fuse Modules for Automotive Fuses, Series 2006

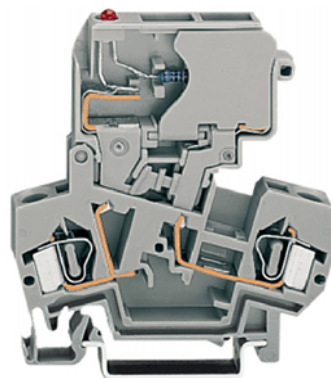


Figure 14: Fuse Modules with Pivotable Fuse Carrier, Series 281

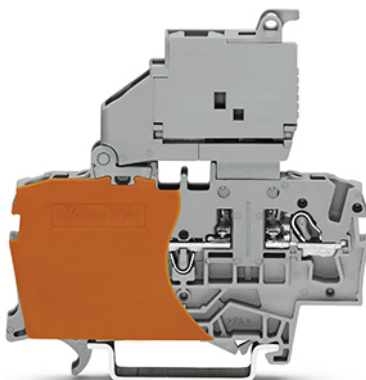


Figure 15: Fuse Modules with Pivotable Fuse Carrier, Series 2002

3.6.5 Supplementary Power Supply Regulations

The WAGO-I/O-SYSTEM 750 can also be used in shipbuilding or offshore and onshore areas of work (e. g. working platforms, loading plants). This is demonstrated by complying with the standards of influential classification companies such as Germanischer Lloyd and Lloyds Register.

Filter modules for 24 V supply are required for the certified operation of the system.

Table 8: Filter Modules for 24 V Supply

Order No.	Name	Description
750-626	Supply Filter	Filter module for system supply and field supply (24 V, 0 V), i. e. for fieldbus coupler/controller and bus power supply (750-613)
750-624	Supply Filter	Filter module for the 24 V field supply (750-602, 750-601, 750-610)

Therefore, the following power supply concept must be absolutely complied with.

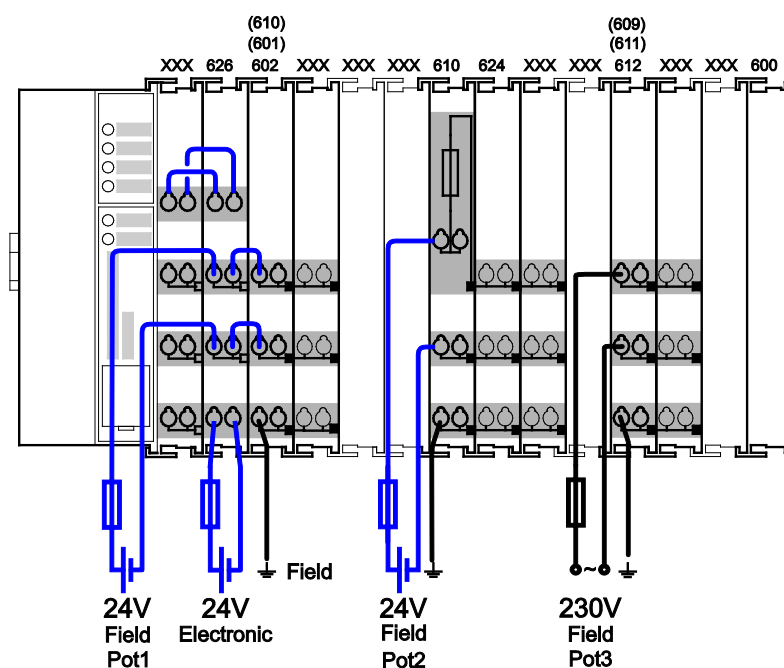


Figure 16: Power Supply Concept

Note



Use a supply module for equipotential bonding!

Use an additional 750-601/ 602/ 610 Supply Module behind the 750-626 Filter Module if you want to use the lower power jumper contact for equipotential bonding, e.g., between shielded connections and require an additional tap for this potential.

3.6.6 Supply Example

Note



The system supply and the field supply shall be separated!

You should separate the system supply and the field supply in order to ensure bus operation in the event of a short-circuit on the actuator side.

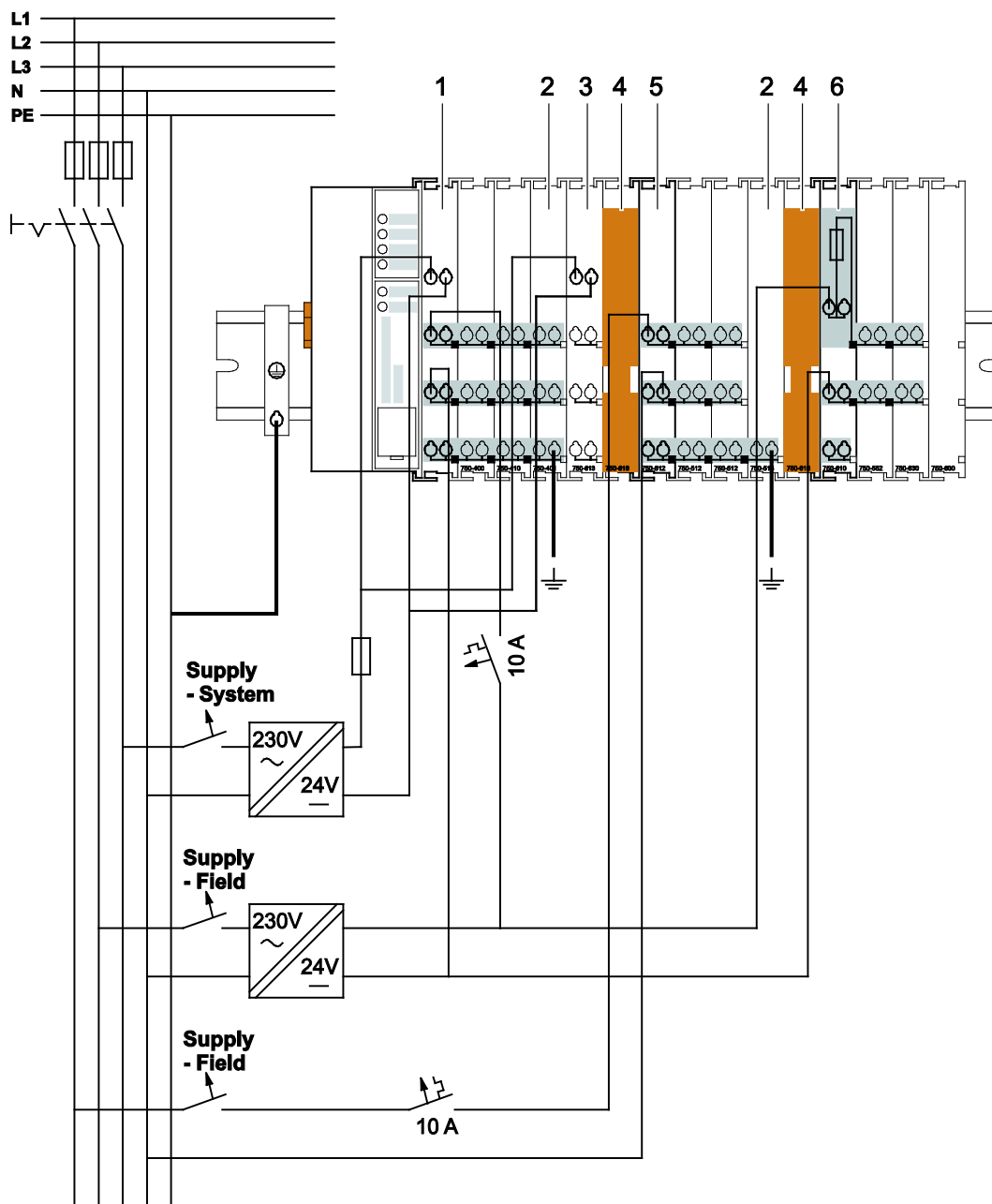


Figure 17: Supply Example for Standard Couplers/Controllers

Table 9: Legend for Figure “Supply Example for Fieldbus Coupler/Controller”

Pos.	Description
1	Power Supply on fieldbus coupler/controller via external Supply Module
2	Power Supply with optional ground
3	Internal System Supply Module
4	Separation module recommended
5	Supply Module passive
6	Supply Module with fuse carrier/diagnostics

3.6.7 Power Supply Unit

The WAGO-I/O-SYSTEM 750 requires a 24 VDC voltage (system supply).

Note



Recommendation

A stable power supply cannot always be assumed everywhere. Therefore, you should use regulated power supplies to ensure the quality of the supply voltage.

For brief voltage dips, a buffer (200 µF per 1 A load current) must be provided.

Note



Buffer for system power supply!

The system power supply must be buffered to bridge power outages. As the power demand depends on the respective node configuration, buffering is not implemented internally.

To achieve power outages of 1 ms to 10 ms according to IEC61131-2, determine the buffering appropriate for your node configuration and structure it as an external circuit.

The power demand must be determined individually depending on the entry point of the field supply. All loads through field devices and I/O modules must be taken into account. The field supply also impacts the I/O modules because the input and output drivers of some I/O modules require the voltage of the field supply.

Note



System and field supply must be isolated!

The system supply and field supply must be isolated to ensure bus operation in the event of short circuits on the actuator side.

Information



Power supply units are available in the eShop.

You can find suitable power supply units, e. g. from the EPSITRON series, in the eShop on www.wago.com.

3.7 Grounding

3.7.1 Grounding the DIN Rail

3.7.1.1 Framework Assembly

When setting up the framework, the carrier rail must be screwed together with the electrically conducting cabinet or housing frame. The framework or the housing must be grounded. The electrical connection is established via the screw. Thus, the carrier rail is grounded.



DANGER

Ensure sufficient grounding is provided!

You must take care to ensure the flawless electrical connection between the carrier rail and the frame or housing in order to guarantee sufficient grounding.

3.7.1.2 Insulated Assembly

Insulated assembly has been achieved when there is constructively no direct ohmic contact between the cabinet frame or machine parts and the carrier rail. Here, the earth ground must be set up via an electrical conductor in accordance with valid national safety regulations.



Note

Recommendation

The optimal setup is a metallic assembly plate with grounding connection which is electrically conductive linked to the carrier rail.

The separate grounding of the carrier rail can be easily set up with the aid of the WAGO ground wire terminals.

Table 10: WAGO Ground Wire Terminals

Order No.	Description
283-609	1-conductor ground (earth) terminal block make an automatic contact to the carrier rail; conductor cross section: 0.2 mm ² ... 16 mm ² Note: Also order the end and intermediate plate (283-320).

3.7.2 Grounding Function

The grounding function increases the resistance against electro-magnetic interferences. Some components in the I/O system have a carrier rail contact that dissipates electro-magnetic interferences to the carrier rail.

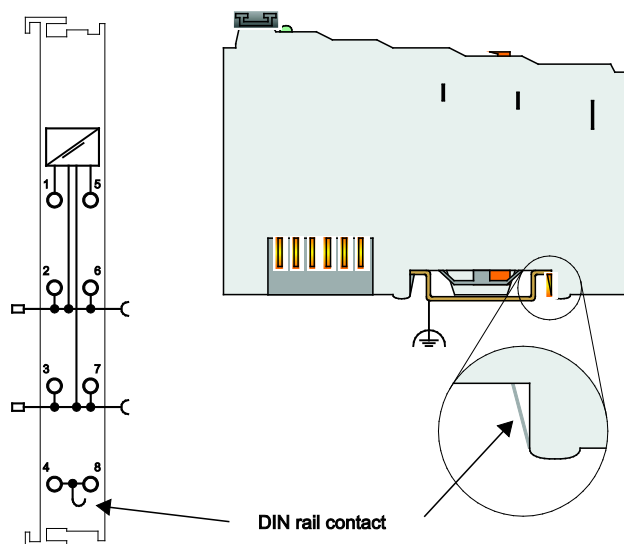


Figure 18: Carrier Rail Contact (Example)



DANGER

Ensure sufficient grounding is provided!

You must take care to ensure the direct electrical connection between the carrier rail contact and the carrier rail.

The carrier rail must be grounded.

For information on carrier rail properties, see section “Mounting” > ... > “Carrier Rail Properties”.

The bottom CAGE CLAMP[®] connectors of the supply modules enable optional connection of a field-side functional ground. This potential is made available to the I/O module arranged on the right through the spring-loaded contact of the three power contacts. Some I/O modules are equipped with a knife-edge contact that taps this potential. This forms a potential group with regard to functional ground with the I/O module arranged on the left.

3.8 Shielding

3.8.1 General

Use of shielded cables reduces electromagnetic interference and thus increases signal quality. Measurement errors, data transmission errors and interference due to excessive voltage can be prevented.

Note



Connect the cable shield to the ground potential!

Integrated shielding is mandatory to meet the technical specifications in regards to measuring accuracy. Connect the cable shield and ground potential at the inlet to the cabinet or housing. This allows induced interference to dissipate and to be kept away from devices in the cabinet or housing.

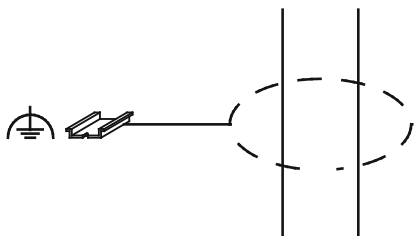


Figure 19: Cable Shield at Ground Potential

Note



Improve shielding performance by placing the shield over a large area!

Higher shielding performance is achieved via low-impedance connection between shield and ground. For this purpose, connect the shield over a large surface area, e.g., WAGO shield connecting system. This is especially recommended for large-scale systems where equalizing current or high impulse-type currents caused by atmospheric discharge may occur.

Note



Keep data and signal lines away from sources of interference!

Route data and signal lines separately from all high voltage cables and other sources of high electromagnetic emission (e.g., frequency converter or drives).

3.8.2 Bus Cables

The shielding of bus lines is described in the respective configuration guidelines and standards of the bus system.

3.8.3 Shielded Signal Lines



Note

Use shielded signal lines!

Always use shielded signal lines for analog signals and I/O modules which are equipped with shield clamps. Only then you can ensure that the accuracy and interference immunity specified for the respective I/O module can be achieved even in the presence of interference acting on the signal cable.

On some WAGO devices you can directly clamp the shield. For all other devices use the WAGO shield connecting system.

3.8.4 WAGO Shield Connecting System

The series 790 WAGO shield connecting system consists of shield clamping saddles, busbars and various mounting carriers. These components can be used to achieve many different configurations.

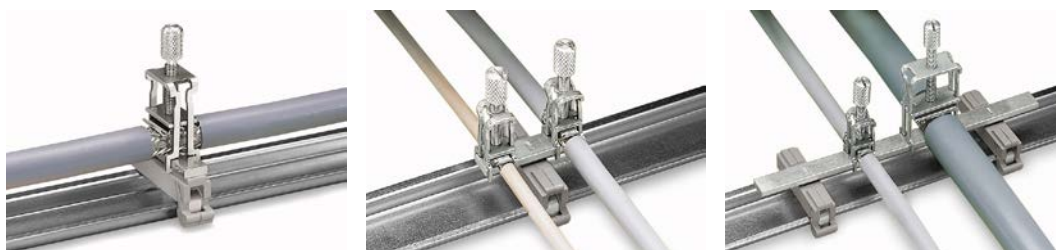


Figure 20: Examples of the WAGO Shield Connecting System

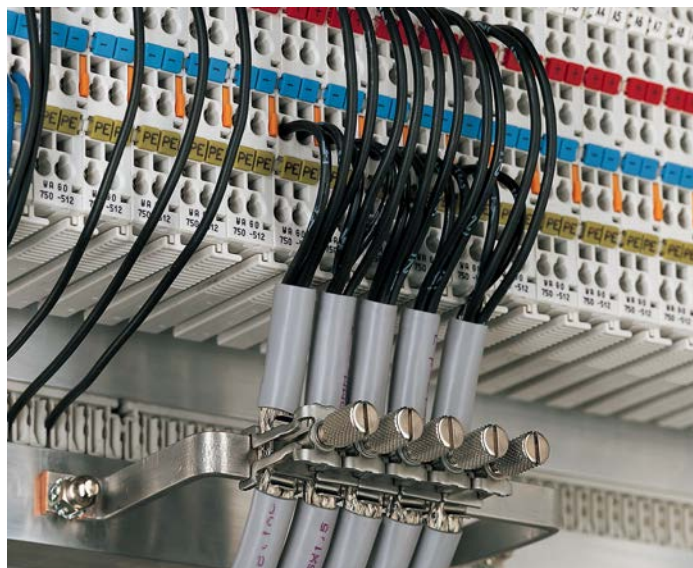


Figure 21: Application of the WAGO Shield Connecting System

4 Device Description

The 750-831 BACnet/IP Controller connects the WAGO-I/O-SYSTEM with the BACnet protocol. The 750-831 Controller complies with the BACnet device profile "BACnet Building Controller" B-BC in accordance with DIN EN ISO 16484-5 and has 3 functions available internally:

1. **Native Server:** For each channel, appropriate BACnet objects are generated automatically for the digital, analog input and output modules that are connected to the controller.
2. **Application Server:** Other supported BACnet objects can be created via the IEC 61131-3 programming environment and made available to a BACnet network.
3. **Application-Client:** Using the client functionality, objects and their properties can be accessed by other BACnet devices.

The BACnet/IP Controller can be used for applications in building automation.

Two RJ-45 ETHERNET interfaces and an integrated switch allow the fieldbus to be wired in a line topology and allow access to BACnet/IP networks. This eliminates additional network devices, such as switches or hubs.

The DIP switch configures the last byte of the IP address and may be used for IP address assignment (e.g. via BootP).

In the fieldbus controller, all input signals from the sensors are combined. After connecting the fieldbus controller, the fieldbus controller determines which I/O modules are on the node and creates a local process image from these. Analog and specialty module data is sent via words and/or bytes; digital data is grouped bit-by-bit.

The local process image is divided into two data zones containing the data received and the data to be sent.

The data of the analog modules is mapped first into the process image. The modules are mapped in the order of their physical position after the controller.

The bits of the digital modules are combined into words and then mapped after the analog ones in the process image. If the number of digital I/Os is greater than 16 bits, the Fieldbus Controller automatically begins a new word.

According to IEC 61131-3 programming, data processing occurs in the PFC. The process results can be output directly on sensors/actuators or transmitted via fieldbus to the higher-order controller.

The fieldbus connection consists of two ports (RJ-45). An ETHERNET switch integrated in the fieldbus coupler/controller, which is operated in the store and forward mode, connects those fieldbus ports with the CPU.

Both ports support:

- 10BASE-T / 100BASE-TX
- Full / Half duplex
- Autonegotiation
- Auto-MDI(X)

WAGO-I/O-PRO creates application programs that adhere to IEC 61131-3. CODESYS by 3S (the standard programming system) serves as the basis of WAGO-I/O-PRO, which was expanded specifically with the target files for all WAGO controllers.



Information

Compatibility with IEC-61131-3 programming software!

The compatibility between your fieldbus controller and the IEC 61131-3 programming software used depends on the fieldbus controller's firmware version and the version of the programming software.

A list of recommended combinations can be found at www.wago.com.

Use the search function (search for "Compatibility Notes").

The fieldbus controller has 1024 KB program memory, 1024 KB data memory and 28 KB retentive memory available for the IEC 61131-3 programming.

Depending on configuration the user can access all fieldbus and I/O data.

In order to send process data via ETHERNET, the controller supports a series of network protocols.

The MODBUS/TCP protocol and the BACnet/IP protocol are implemented for exchanging process data. Both of these communication protocols can be used either simultaneously or alternately. Write access from the fieldbus controller to the I/O modules is defined in a configuration file called "EA-config.xml" via the I/O Configurator in WAGO-I/O-PRO.

Any combination of supported digital, analog and complex I/O modules of the WAGO-I/O-SYSTEM 750 can be used here.



Note

Non-supported I/O modules for native access via the BACnet protocol!

Note that the following I/O modules are not supported for BACnet access:

- 750-418 (digital input module, 2-channel, 24 VDC, 3.0 ms)
- 750-439 (digital input module, 8-channel, NAMUR Ex i)
- 750-1425 (digital input module, 8-channel, PTC)
- 750-645 (Special Modules, 2-channel, Vibration Velocity/Bearing Condition Monitoring VIB I/O)

A CODESYS application is required in order to use these modules.

Start-up and configuration of the BACnet/IP Controller is performed using the WAGO BACnet Configurator.

For the management and diagnosis of the system, the HTTP, SNTP and SNMP protocols are available.

For the data transfer via ETHERNET the FTP is available.

For the assignment of the IP address in the network BootP can be used.

The user can program clients and servers via an internal socket-API for all transport protocols (TCP, UDP, etc.) with functional modules. Library functions are available for function expansion.

With the IEC 61131-3 library "SysLibRTC.lib," for example, a buffered real-time clock with date, time (1-second resolution), alarm functions and a timer is incorporated. This clock is supplied with auxiliary power during a power failure.

This controller is based on a 32-bit CPU with multitasking capabilities, allowing several programs to be executed in a near-simultaneous manner.

Start-up and configuration of the BACnet/IP Controller is performed using the WAGO BACnet Configurator.

The controller has an internal server for the configuration and administration of the system.

The internal file system supports two drives, a 4,5MB remanent memory, and a pluggable memory card (SD card).

Note



Memory card is not included in the scope of delivery!

Note, the BACnet/IP Controller is delivered without memory card.

To use a memory card, you must order one separately. The BACnet/IP Controller can also be operated without memory card expansion, the use of a memory card is optional.

Note



Only use recommended memory cards!

Use only the SD memory card available from WAGO (order no. 758-879/000-001) since it is suitable for industrial applications under difficult environmental conditions and for use in the BACnet/IP Controller.

The compatibility to other storage media available in trade cannot be ensured.

By default, the controller's built-in HTML pages contain information on the configuration and status of the PFC, and can be read using a normal web browser. In addition, a file system is implemented that allows you to store custom HTML

pages in the controller using FTP download or to store your own HTML pages or call up programs directly.

Information



Additional Information about current Software

To get the current software version for programming and configuring the 750-831 BACnet/IP Controller, go to our website at: <http://www.wago.com> in the Documentation area → WAGO Software → WAGO-I/O-PRO.

Information



Additional Information about BACnet-Configurator

You can find the BACnet Configurator on the internet on the website: <http://www.wago.com> in the Downloads area → BACnet Configurator. The documentation for the BACnet Configurator can be found under: <http://www.wago.com> in the Documentation area → WAGO Software → WAGO BACnet Configurator.

4.1 View

The view below shows the three parts of the device:

- The fieldbus connection is on the left side.
- LEDs for operation status, bus communication, error messages and diagnostics, as well as the service interface are in the middle area.
- The right side shows the power supply unit for the system supply and for the field supply of the attached I/O modules via power jumper contacts. LEDs show the status of the operating voltage for the system and field supply (jumper contacts).

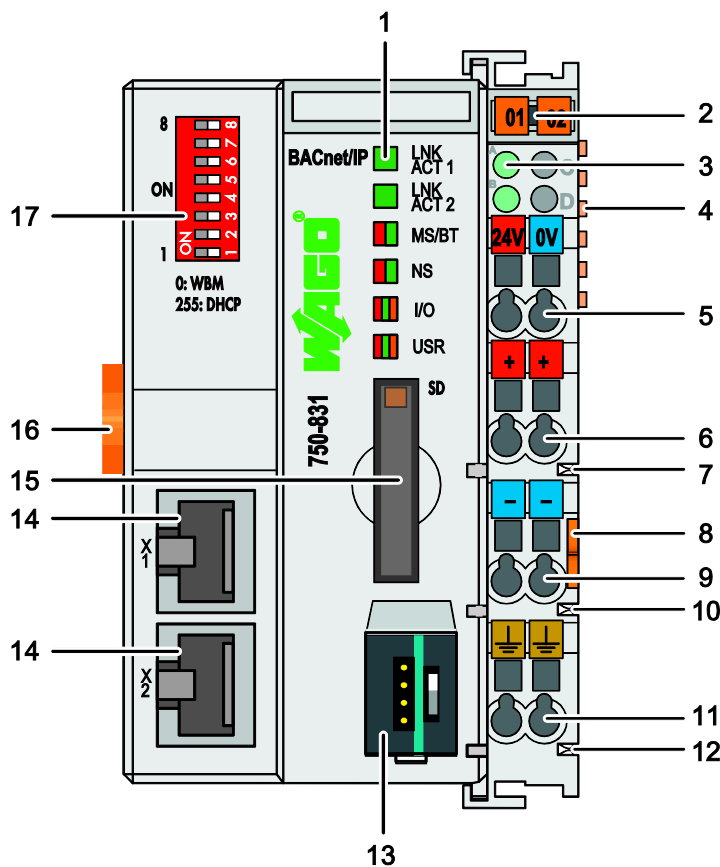


Figure 22: View BACnet/IP Controller

Table 11: Legend for Figure “View BACnet/IP Controller 750-831”

Pos.	Designation	Meaning	Details see Section
1	LNK ACT 1, LNK ACT 2, MS/BT, NS, I/O, USR	Status LEDs Fieldbus	"Device Description" > "Display Elements"
2	---	Group marking carrier (retractable) with additional marking possibility on two miniature WSB markers	---
3	A, B or C	Status LED's System/Field Supply	"Device Description" > "Display Elements"
4	---	Data Contacts	"Connect Devices" > "Data Contacts/Internal Bus"
5	24 V, 0 V	CAGE CLAMP® Connections System Supply	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
6	+	CAGE CLAMP® Connections Field Supply 24 VDC	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
7	---	Power Jumper Contact 24 VDC	"Connect Devices" > "Power Contacts/Field Supply"
8	---	Unlocking Lug	"Mounting" > "Inserting and Removing Devices"
9	-	CAGE CLAMP® Connections Field Supply 0 V	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
10	---	Power Jumper Contact 0 V	"Connect Devices" > "Power Contacts/Field Supply"
11	(Ground)	CAGE CLAMP® Connections Field Supply (Ground)	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
12	---	Power Jumper Contact (Ground)	"Connect Devices" > "Power Contacts/Field Supply"
13	---	Service Interface (open flap)	"Device Description" > "Operating Elements"
14	X1, X2	Fieldbus connection 2 x RJ-45 as 2-Port Switch	"Device Description" > "Connectors"
15	---	SD card slot with cover lid	„Device Description“ > „Operating Elements“
16	---	Locking Disc	"Mounting" > "Inserting and Removing Devices"
17	---	Address Selection Switch	"Device Description" > "Operating Elements"

4.2 Connectors

4.2.1 Device Supply

The device is powered via terminal blocks with CAGE CLAMP® connections.

The device supply generates the necessary voltage to power the electronics of the device and the internal electronics of the connected I/O modules.

The fieldbus interface is galvanically separated to the electrical potential of the device.

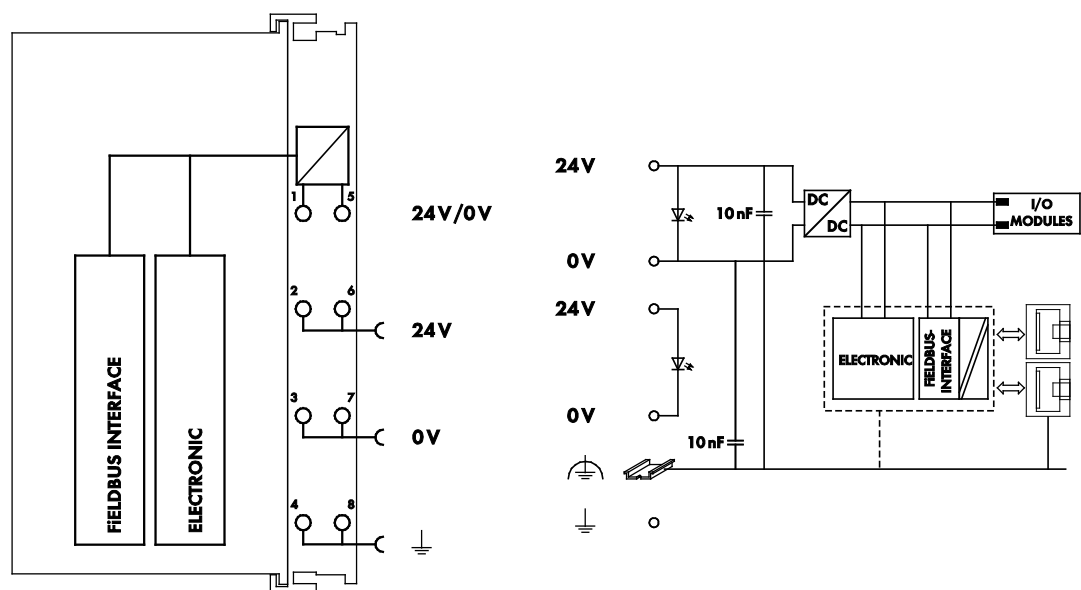


Figure 23: Device Supply

4.2.2 Fieldbus Connection

The connection to the ETHERNET based fieldbuses is made via two RJ-45 plugs (also called “Western plugs”), which are connected to the fieldbus controller via an integrated switch.

The integrated switch works in store-and-forward operation and for each port, supports the transmission speeds 10/100 Mbit as well as the transmission modes full and half-duplex.

The wiring of these plugs corresponds to the specifications for 100BaseTX, which prescribes a category 5 twisted pair cable as the connecting cable. Cable types S-UTP (Screened Unshielded Twisted Pair) and STP (Shielded Twisted Pair) with a maximum segment length of 100 m (approximately 328.08 feet) can be used.

The socket is arranged physically lower, allowing the coupler to fit in an 80 mm high enclosure after plug connection.

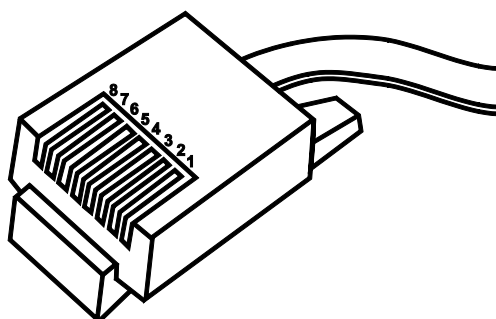


Figure 24: RJ-45 Connector

Table 12: RJ-45 Connector and RJ-45 Connector Configuration

Contact	Signal	
1	TD +	Transmit +
2	TD -	Transmit -
3	RD +	Receive +
4		free
5		free
6	RD -	Receive -
7		free
8		free

NOTICE

Do not use in telecommunication circuits!

Only use devices equipped with ETHERNET or RJ-45 connectors in LANs.
Never connect these devices with telecommunication networks.

4.3 Display Elements

The operating condition of the fieldbus controller or the node is displayed with the help of illuminated indicators in the form of light-emitting diodes (LEDs). The LED information is routed to the top of the case by light fibres. In some cases, these are multi-colored (red, green or red/green (=orange)).

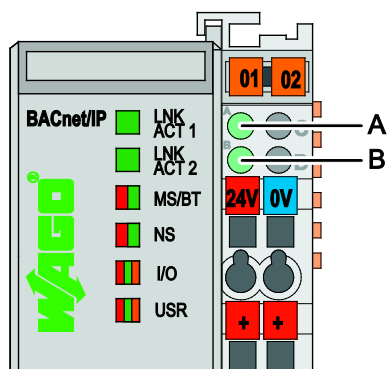


Figure 25: Display Elements

For the diagnostics of the different domains fieldbus, node and supply voltage, the LEDs can be divided into three groups:

Table 13: Display Elements Fieldbus Status

LED	Color	Meaning
LNK ACT 1	green	indicates a connection to the physical network at port 1
LNK ACT 2	green	indicates a connection to the physical network at port 2
MS/BT	green/ green blinking/ red/ red blinking	indicates the node or BACnet status
NS	red/green	indicates the network status

Table 14: Display Elements Node Status

LED	Color	Meaning
I/O	red/green/ orange	Indicates the operation of the node and signals via a blink code faults encountered.
USR	red/green/ orange	indicates the state, which is programmed in the user program

Table 15: Display Elements Supply Voltage

LED	Color	Meaning
A	green	indicates the status of the operating voltage – system
B	green	indicates the status of the operating voltage – power jumper contacts



Information

More information about the LED Signaling

Read the detailed description for the evaluation of the displayed LED state in the section “Diagnostics” > ... > “LED Signaling”.

4.4 Operating Elements

4.4.1 Service Interface

The service interface is located behind the flap.

It is used for the communication with WAGO-I/O-CHECK, WAGO-I/O-PRO and for update the firmware.

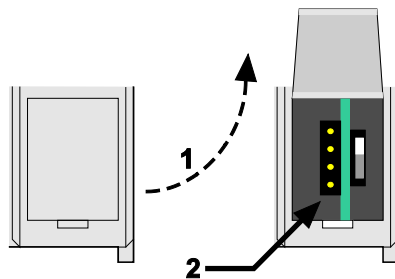


Figure 26: Service Interface (Closed and Opened Flap)

Table 16: Legend for Figure “Service Interface (Closed and Opened Flap)”

Number	Description
1	Open closed flap
2	View Service Interface

NOTICE

Device must be de-energized!

To prevent damage to the device, unplug and plug in the communication cable only when the device is de-energized!

The connection to the 4-pin header under the cover flap can be realized via the communication cables with the item numbers 750-920 and 750-923 or via the WAGO radio adapter with the item number 750-921.

4.4.2 Mode Selector Switch

The mode selector switch is located behind the cover flap.

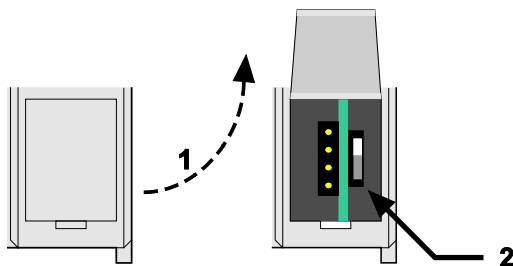


Figure 27: Mode Selector Switch (Closed and Open Damper of the Service Port)

Table 17: Legend for Figure „Mode Selector Switch“

Number	Description
1	Open the damper
2	Mode selector switch

The mode selector switch determines the loading, starting and stopping of the PLC-application by the fieldbus controller. This multifunction sliding switch features 3 slide lock positions and a push-button function.

The sliding switch is designed for a number of operations in compliance with EN61131T2.

NOTICE

Property damages due to set outputs!

Please note that set outputs remain set, when you switch the operating switch from “RUN” to “STOP” during the current operation. Since the program is no longer processed, software-related switch offs, i.e. by initiators, are ineffective. Therefore, program or define all outputs, so that these switch to a safe mode at a program stop.

Note



Defining the outputs for a program stop!

In order to switch the outputs to a safe mode at the program stop, define the status of the outputs at “STOP”.

1. For this, open in the web-based Management System (WBM) a website via the “PLC Settings” link, on which you can define the function **Process image - Set outputs to zero, if user program is stopped**.
2. Now activate this function by placing a check mark in the control box, then all outputs are set to zero, if this function is not activated, the outputs remain at the last current value.



Note

The mode selector switch position does not affect software start/stop!

The position of the operating mode switch does not prevent the starting and stopping of the PFC application from WAGO-I/O-PRO.

One of the following functions is active, depending on which of the three static positions — “top”, “center” or “bottom” — the switch is located at when energized or during a hardware or software reset:

Table 18: Mode Selector Switch Positions, Static Positions on PowerOn/Reset

Positions for the mode selector switch	Function
“Top” position	“RUN” – activate program processing, Boot project (if available) is started
“Center” position	“STOP” – stop program processing, PFC application is stopped
“Bottom” position	Do not use. This position is not relevant for the user.

The fieldbus controller performs the following functions if the switch’s position is changed during operation:

Table 19: Mode Selector Switch Positions, Dynamic Positions During Ongoing Operation

Position change for the mode selector switch	Function
From the top to the center position	“STOP” – stop program processing, PFC application is stopped
From the center to the top position	“RUN” – activate program processing, Boot project (if available) is started
From the center to the bottom position	No reaction. The bootstrap loader is started after PowerOn/Reset
From the bottom to the center position	No reaction.
Press down (e.g., using a screwdriver)	Hardware reset. All outputs are reset; variables are set to 0, FALSE, or to an initial value. Retain variables or markers are not changed. A hardware reset can be executed on STOP or on RUN at any position of the mode selector switch! Fieldbus coupler restart.

The operating mode is changed internally at the end of a PFC cycle.

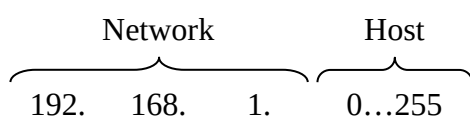
4.4.3 Address Selection Switch



Figure 28: Address Selection Switch (for example setting “0”)

The 8-pole DIP switch is used to set the IP address and to select the protocol for setting the IP address.

The IP address is composed of a network component and a host component.



The network component is configurable and in delivery status 192.168.1.
The host component is set via the DIP switch.

The individual DIP switches 1 to 8 represent a bit mask corresponding to the values 0 to 255. Switch 1 corresponds to the lowest bit (2^0) and switch 8 corresponds to the highest bit (2^7).

Table 20: Meaning of DIP Switch Positions

Host Component of the IP Address	Meaning
0 (WBM)	The IP configuration set in the WBM on the "Port" page is used (default BootP, but can be changed to DHCP or EEPROM).
1 ... 254	Host component by switch position Example of setting the DIP switch: For the IP address 192.168.1.33, the 1st and 6th switches must be set: 2^0 (1st switch) + 2^5 (6th switch) = 33
255 (DHCP)	The DHCP protocol is used to configure the IP parameters.

4.4.4 Memory Card Slot

The memory card slot accommodates a secure digital memory card (or SD card for short) or an SD card with a higher storage capacity (SD high capacity, or SDHC for short).

Note



Only use recommended memory cards!

Use only the SD memory card available from WAGO (order no. 758-879/000-001) since it is suitable for industrial applications under difficult environmental conditions and for use in the fieldbus controller.

The compatibility to other storage media available in trade cannot be ensured.

The memory card slot has a transparent protective flap that is folded upwards to open it.

Note



Memory card is not included in the scope of delivery!

Note, the fieldbus controller is delivered without memory card.

To use a memory card, you must order one separately. The fieldbus controller can also be operated without memory card expansion, the use of a memory card is optional.

Information



Additional information about the memory card

For technical information about the memory card, see the data sheet for the SD memory card (order no. 758-879/000-001).

The data sheet is on the Internet at: www.wago.com

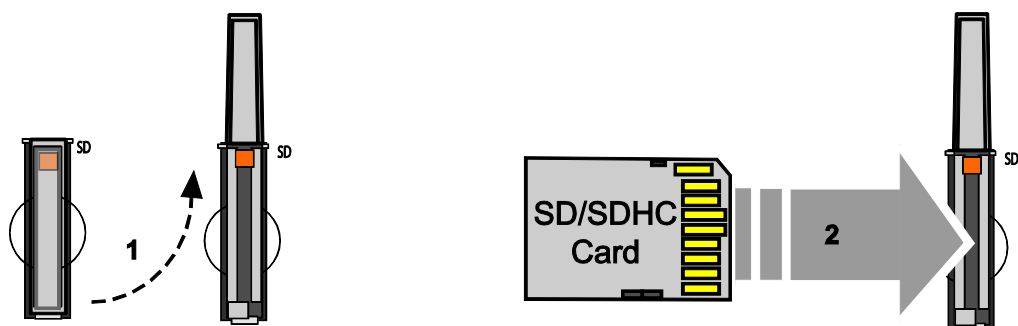


Figure 29: Opening the Memory Card Slot, Inserting an SD/SDHC Memory Card

4.4.4.1 Inserting a Memory Card

1. Use an activation tool or a screwdriver to open the transparent cover flap by folding it upwards.
2. Hold the memory card so that the contacts are visible on the right side and the sloping edge is towards the bottom, as depicted in the figure above.
3. Insert the memory card into the slot of the fieldbus controller.
4. Push the memory card all the way in. When you let go, the memory card will move back a little and then snap in.
5. Push the cover flap back in by folding it downwards until it snaps in.

4.4.4.2 Removing the Memory Card

1. Use an activation tool or a screwdriver to open the transparent cover flap by folding it upwards.
2. To remove the memory card, first press it into the slot.
This loosens the mechanical locking
3. As soon as you let go of the memory card, the memory card is pushed out a bit and you can remove it.
4. Push the cover flap back in by folding it downwards until it snaps in.

4.5 Technical Data

4.5.1 Device Data

Table 21: Technical Data – Device Data

Width	62 mm
Height (from upper-edge of DIN 35)	65 mm
Length	100 mm
Weight	approx. 164 g
Degree of protection	IP 20

4.5.2 System Data

Table 22: Technical Data – System Data

Number of controllers connected to Master	Limited by ETHERNET specification
Transmission medium	Twisted Pair S/UTP, STP 100 Ω Cat 5
Max. length of fieldbus segment	100 m acc. to IEEE-802.3 specification
Network length _{max}	acc. to IEEE-802.3 specification
Transfer rate	10/100 Mbit/s
Bus coupler connection	2 x RJ-45
Protocols	BACnet/IP, MODBUS/TCP, HTTP, BootP, DHCP, DNS, SNTP, FTP, SNMP
Transmission performance	Class D acc. to EN 50173
BACnet implementation acc. to	ISO 16484-5:2010 BACnet protocol revision 7
BACnet Device Profile	BACnet Building Controller (B-BC)
Programming	WAGO-I/O-PRO
IEC-61131-3	AWL, KOP, FUP, ST, AS
SD memory card slot	push/push mechanism, cover cap lead-sealable
Memory card type	SD and SDHC up to 32 Gbyte *)
Max. number of socket links	3 HTTP, 30 MODBUS/TCP, 10 FTP, 2 SNMP, 5**) for IEC-61131-3 program, 2 for WAGO-I/O-PRO
Powerfail RTC Buffer	at least 6 days***)
Number of I/O modules with bus extension	64 99
Configuration	via PC
Program memory	1024 kByte
Data memory	1024 kByte
Non-volatile memory (retain)	28 kByte

- *) Use only the SD memory card available from WAGO (order no. 758-879/000-001) since it is suitable for industrial applications under difficult environmental conditions and for use in the fieldbus controller. The compatibility to other storage media available in trade cannot be ensured.
- **) Using the Ethernet.lib, when the SysLibSocket.lib is used, there are 143.
- ***) This value is valid for brand-new devices with an ambient temperature of 25 °C.
The guaranteed buffer time for the real time clock is reduced with rising temperature and operating time.

4.5.3 Supply

Table 23: Technical Data – Supply

Voltage via power jumper contacts	24 V DC (-25 % ... +30 %)
Current via power jumper contacts _{max.}	10 A DC
Power failure time acc. IEC 61131-2	Depending on external buffering
Input current _{typ.} at rated load (24 V)	500 mA
Efficiency of power supply _{typ.} at nominal load	90 %
Internal current consumption	450 mA at 5 V
Total current for I/O modules	1700 mA at 5 V
Isolation	500 V system/supply



Note

Buffer for system power supply!

The system power supply must be buffered to bridge power outages. As the power demand depends on the respective node configuration, buffering is not implemented internally.

To achieve power outages of 1 ms to 10 ms according to IEC61131-2, determine the buffering appropriate for your node configuration and structure it as an external circuit.

4.5.4 Fieldbus MODBUS/TCP

Table 24: Technical Data – Fieldbus MODBUS/TCP

Input process image _{max}	1020 words
Output process image _{max}	1020 words

4.5.5 Accessories

Table 25: Technical Data – Accessories

Miniature WSB Quick marking system	
PC Software	WAGO BACnet Configurator
WAGO-I/O-PRO	759-333
SD Memory Card	758-879/000-001

4.5.6 Connection Type

Table 26: Technical Data – Field Wiring

Wire connection	CAGE CLAMP®
Cross section	0.08 mm² ... 2.5 mm², AWG 28 ... 14
Stripped lengths	8 mm ... 9 mm / 0.33 in

Table 27: Technical Data – Power Jumper Contacts

Power jumper contacts	Spring contact, self-cleaning
Voltage drop at I_{max} .	< 1 V/64 modules

Table 28: Technical Data – Data Contacts

Data contacts	Slide contact, hard gold plated, self-cleaning

4.5.7 Climatic Environmental Conditions

Table 29: Technical Data – Climatic Environmental Conditions

Operating temperature range	0 °C ... 55 °C
Operating temperature range for components with extended temperature range (750-xxx/025-xxx)	-20 °C ... +60 °C
Storage temperature range	-25 °C ... +85 °C
Storage temperature range for components with extended temperature range (750-xxx/025-xxx)	-40 °C ... +85 °C
Relative humidity	Max. 5 % ... 95 % without condensation
Resistance to harmful substances	Acc. to IEC 60068-2-42 and IEC 60068-2-43
Maximum pollutant concentration at relative humidity < 75 %	SO ₂ ≤ 25 ppm H ₂ S ≤ 10 ppm
Special conditions	Ensure that additional measures for components are taken, which are used in an environment involving: – dust, caustic vapors or gases – ionizing radiation

NOTICE**Reduced buffer time at high storage temperature!**

Ensure that the storage of devices with a real time clock at high temperatures leads to a reduced buffer time for the real time clock.

4.5.8 Mechanical Strength

Table 30: Technical Data – Mechanical Strength

Vibration resistance	acc. to IEC 60068-2-6 Comment to the vibration resistance: a) Type of oscillation: sweep with a rate of change of 1 octave per minute $10\text{ Hz} \leq f < 57\text{ Hz}$, const. Amplitude 0,075 mm $57\text{ Hz} \leq f < 150\text{ Hz}$, const. Acceleration 1 g b) Period of oscillation: 10 sweep per axis in each of the 3 vertical axes
Shock resistance	acc. to IEC 60068-2-27 Comment to the shock resistance: a) Type of impulse: half sinusoidal b) Intensity of impulse: 15 g peak value, 11 ms maintenance time c) Route of impulse: 3 impulses in each pos. And neg. direction of the 3 vertical axes of the test object, this means 18 impulses in all.
Free fall	acc. IEC 60068-2-32 $\leq 1\text{ m}$ (module in original packing)

4.6 Approvals



Information

More information about approvals.

Detailed references to the approvals are listed in the document “Overview Approvals **WAGO-I/O-SYSTEM 750**”, which you can find via the internet under: www.wago.com > SERVICES > DOWNLOADS > Additional documentation and information on automation products > WAGO-I/O-SYSTEM 750 > System Description.

The following approvals have been granted to the basic version of the fieldbus coupler/controller “BACnet/IP Controller” (750-831):



Conformity Marking



cUL_{US}

UL508



Korea Certification

MSIP-REM-W43-PFC750

The following BACnet approvals have been granted to the respective software versions (SW):

Table 31: BACnet Approvals

BACnet Approvals	750-831	750-831/000-002
BACnet conformance test acc. ISO 16484-5:2014	Passed SW 01.02.29(09)	Passed SW 01.02.29(09)
 BTL Certification (Merger of WSPCert certification and BTL Listing from 2017)	Pending *	Pending *
AMEV attestation AMEV Profile AS-B	Pending *	Pending *

* Status as of 1/2017

Further certifications are currently in preparation.

4.7 Standards and Guidelines

750-831 meets the following requirements on emission and immunity of interference:

EMC CE-Immunity to interference EN 61000-6-2

EMC CE-Emission of interference EN 61000-6-3

4.8 BACnet-Building-Controller (B-BC)

The BACnet Standard 135-2010 describes eight BACnet device profiles. Any device that implements all the required BACnet capabilities for a particular device type and interoperability area may claim to be a device of that particular device profile. Devices may also provide additional capabilities. All supported capabilities of the controller must be indicated in the protocol implementation confirmation (PICS). This is a document that represents and compares the capabilities of the device.



Information

Additional Information

You can receive background information on the BACnet technology, for example on the topics “Interoperability Areas” and “Device Profiles” in the general BACnet section “BACnet”.

The 750-831 WAGO BACnet/IP Controller represents the device profile of the BACnet Building Controllers (B-BC). As such, the controller serves as a programmable automation system that can take over a multitude of different building automation and control tasks. It enables the specification of the following:

Data Sharing

- Ability to provide the values of any of its BACnet objects
- Ability to retrieve the values of BACnet objects from other devices
- Ability to allow modification of some or all of its BACnet objects by another device
- Ability to modify some BACnet objects in other devices

Alarm and Event Management

- Generation of alarm/event notifications and the ability to direct them to recipients
- Maintain a list of unacknowledged alarms/events
- Notifying other recipients that the acknowledgment has been received
- Adjustment of alarm and event parameters, scheduling
- Ability to schedule output actions, both in the local device and in other devices, both binary and analog, based on date and time

Trending

- Collection and delivery of (time, value) pairs

Device und Network Management

- Ability to respond to queries about its status
- Ability to respond to requests for information about any of its objects
- Ability to respond to communication control messages
- Ability to synchronize its internal clock upon request
- Ability to perform re-initialization upon request
- Ability to upload its configuration and allow it to be subsequently restored
- Commands for half routers for establishing and breaking off connections

The following table shows the minimum requirement for the BIBBs for the B-BC in general as well as additional BIBBs implemented by the BACnet/IP Controller.

Table 32: BIBBs of the B-BC

IOB	BIBBs of the B-BC ¹ (Minimum requirements)	Additional BIBBs of the BACnet/IP Controller ²
Data Sharing	DS-RP-A,B DS-RPM-A,B DS-WP-A,B DS-WPM-B	DS-COV-A,B DS-COVP-B
Alarm and Event Management	AE-N-I-B AE-ACK-B AE-INFO-B AE-ESUM-B	AE-N-E-B AE-ASUM-B
Scheduling	SCHED-E-B	SCHED-I-B
Trending	T-VMT-I-B T-ATR-B	T-VMT-E-B
Device Management	DM-DDB-A,B DM-DOB-B DM-DCC-B DM-TS-B or DM-UTC-B DM-RD-B DM-BR-B NM-CE-A	DM-R-A DM-R-B DM-LM-B DM-OCD-B

¹ As of BACnet Standard 135-2010

² As of 06/2016 (For currently supported BIBBs of the BACnet/IP Controller, please refer to the “PICS” document at <http://www.wago.com> in the Documentation area.

4.8.1 Data Sharing BIBBs

These BIBBs prescribe the BACnet capabilities required to interoperably perform the data sharing functions.

Data Sharing - ReadProperty-A (DS-RP-A)

The A device is a user of data from device B.

BACnet Service	Requests	Execute
ReadProperty	x	

Data Sharing - ReadProperty-B (DS-RP-B)

The B device is a provider of data to device A.

BACnet Service	Requests	Execute
ReadProperty		x

Data Sharing-ReadPropertyMultiple-A (DS-RPM-A)

The A device is a user of data from device B and requests multiple values at one time.

BACnet Service	Requests	Execute
ReadPropertyMultiple	x	

Data Sharing-ReadPropertyMultiple-B (DS-RPM-B)

The B device is a provider of data to device A and returns multiple values at one time.

BACnet Service	Requests	Execute
ReadPropertyMultiple		x

Data Sharing-WriteProperty-A (DS-WP-A)

The A device sets a value in device B.

BACnet Service	Requests	Execute
WriteProperty	x	

Data Sharing-WriteProperty-B (DS-WP-B)

The B device allows a value to be changed by device A.

BACnet Service	Requests	Execute
WriteProperty		x

Data Sharing-WritePropertyMultiple-B (DS-WPM-B)

The B device allows multiple values to be changed by device A at one time.

BACnet Service	Requests	Execute
WritePropertyMultiple		x

BIBB - Data Sharing-COV-A (DS-COV-A)

Device A is a user of the COV data from device B.

BACnet Service	Requests	Execute
SubscribeCOV	x	
ConfirmedCOVNotification		x
UnconfirmedCOVNotification		x

The support of subscriptions with limited lifetime is necessary; the support of subscriptions with unlimited lifetime is optional.

Data Sharing-COV-B (DS-COV-B)

Device B is a provider of COV data for device A.

BACnet Service	Requests	Execute
SubscribeCOV		x
ConfirmedCOVNotification	x	
UnconfirmedCOVNotification	x	

Devices that comply with DS-COV-B must support at least five simultaneous subscriptions. The support of subscriptions with a limited lifetime is necessary; the support of subscriptions with an unlimited lifetime is optional.

Data Sharing-COV-B (DS-COV-B)

Device B is a provider of COV data for device A.

BACnet Service	Requests	Execute
SubscribeCOVProperty		x
ConfirmedCOVNotification	x	
UnconfirmedCOVNotification	x	

Devices that comply with DS-COV-B must support at least five simultaneous subscriptions. The support of subscriptions with a limited lifetime is necessary; the support of subscriptions with an unlimited lifetime is optional.

4.8.2 Alarm and Event Management BIBBs

These BIBBs prescribe the BACnet capabilities required to interoperably perform the alarm and event management functions.

Alarm and Event-Notification Internal-B (AE-N-I-B)

Device B generates notifications about alarms and other events.

BACnet Service	Requests	Execute
ConfirmedEventNotification	x	
UnconfirmedEventNotification	x	

Devices claiming conformance to AE-N-I-B must also support either Intrinsic or Algorithmic reporting. Any device that supports the generation of event notifications that require operator acknowledgment must support AE-ACK-B.

Alarm and Event-Notification External-B (AE-N-E-B)

Device B contains an Event Enrollment object that monitors values in another device. Device B is capable of generating event notifications for alarm conditions based on value(s) in another device. Devices confirming to the BIBB shall confirm to DS-RP-A, AE-N-I-B, and shall support at least 1 Event Enrollment object. With an Object_Property_Reference property that supports references to properties in objects contained in other devices. Any devices that supports the generation of event notifications that require operator acknowledgment shall support AE-ACK-B and AE-INFO-B. Any devices that supports the generation of TO-FAULT or TO-OFFNORMAL event notifications shall support AE-INFO-B.

Devices that only support Event Enrollment objects that only support generation of CHANGE_OF_LIFE_SAFETY and/or BUFFER_READY notifications shall not claim support for this BIBB.

Alarm and Event-ACK-B (AE-ACK-B)

Device B processes acknowledgments of previously transmitted alarm/event notifications.

BACnet Service	Requests	Execute
AcknowledgeAlarm		x

To support this BIBB the device must also support acknowledgeable alarms.

Alarm and Event-Information-B (AE-INFO-B)

Device B provides event information to device A

BACnet Service	Requests	Execute
GetEventInformation		x

Alarm and Event-Alarm Summary-B (AE-ASUM-B)

BACnet Service	Requests	Execute
GetAlarmSummary		x

Device B provides summaries of alarms to device A.

Alarm and Event-Enrollment Summary-B (AE-ESUM-B)

Device B provides event enrollments to device A.

BACnet Service	Requests	Execute
GetEnrollmentSummary		x

4.8.3 Scheduling BIBBs

These BIBBs prescribe the BACnet capabilities required to interoperably perform the scheduling functions.

Scheduling-Internal-B (SCHED-I-B)

Device B indicates time and data for the scheduling of values of a certain property of certain objects of the device. Each SCHED-I-B compliant device also has at least one Calendar and Schedule Object for the support of the BIBBs DS-RP-B and DS-WP-B. SCHED-I-B compliant devices must also support DM-TS-B and DM-UTC-B.

The Schedule Object must support at least six entries per day. The property List_Of_Object_Property_Reference must support at least one entry. The Schedule Object must support a non-empty property Exception_Schedule. The property Priority_For_Writing of the Schedule Object must be writable.

Scheduling-External-B (SCHED-E-B)

The B device provides date and time scheduling of the values of specific properties of specific objects in other devices. Devices claiming conformance to SCHED-E-B shall also support SCHED-I-B and DS-WP-A. The List_Of_Object_Property_References property shall support references to objects in external devices.

4.8.4 Trending BIBBs

These BIBBs prescribe the BACnet capabilities required to interoperably perform the trend value processing.

Trending-Viewing and Modifying Trends Internal-B (T-VMT-I-B)

The B device collects the trend log data records in an internal buffer. Each device-claiming conformance to T-VMT-I-B must be able to support at least one Trend Log object.

BACnet Service	Requests	Execute
ReadRange		x

Trending-Automated Trend Retrieval-B (T-ATR-B)

The B device notifies the A device that a trending buffer has acquired a predetermined number of data samples using the BUFFER_READY event algorithm either intrinsically in the Trend Log object or algorithmically using an Event Enrollment object.

BACnet Service	Requests	Execute
ConfirmedEventNotification	x	
ReadRange		x

T-ATR-B compliant devices must support the Trend Log Object.

Trending-Viewing and Modifying Trends External-B (T-VMT-E-B)

The device B is capable of trending properties of objects contained in other devices. The B device shall support T-VMT-I-B and DS-RP-A. The Log_Interval and Log_DeviceObjectProperty properties must be writable.

The Trend Log objects must be capable of trending REAL, Unsigned, INTEGER, BOOLEAN, Bit String, Enumerated and NULL values.

4.8.5 Device- und Network-Management-BIBBs

These Device Management BIBBs prescribe the BACnet capabilities required to interoperably perform the device management functions. The network management BIBBs prescribe the BACnet capabilities required to interoperably perform network management functions.

Device Management-Dynamic Device Binding-A (DM-DDB-A)

The A device searches for information on the device properties of other devices and evaluates their notices.

BACnet Service	Requests	Execute
Who-Is	x	
I-Am		x

Device Management-Dynamic Device Binding-B (DM-DDB-B)

The B device provides information about its device properties and responds to requests to identify itself.

BACnet Service	Requests	Execute
Who-Is		x
I-Am	x	

Device Management-Dynamic Object Binding-B (DM-DOB-B)

The B device provides address information about its objects upon request.

BACnet Service	Requests	Execute
Who-Has		x
I-Have	x	

Device Management-DeviceCommunicationControl-B (DM-DCC-B)

The B device responds to communication control exercised by the A device.

BACnet Service	Requests	Execute
DeviceCommunicationControl		x

Support for requests of a limited duration is required, and support for requests of an indefinite duration is optional.

Device Management-Restart-A (DM-R-A)

The A device processes restart notifications.

BACnet Service	Requests	Execute
UnconfirmedCOVNotification		x

Device Management-Restart-B (DM-R-B)

The B device informs the A device(s) each time it restarts.

BACnet Service	Requests	Execute
UnconfirmedCOVNotification	x	

Devices claiming conformance to DM-R-B shall support the Last_Restart_Reason, Restart_Notification_Recipients, and Time_Of_Device_Restart properties of the Device object.

Device Management-TimeSynchronization-B (DM-TS-B)

The B device interprets time synchronization messages from the A device.

BACnet Service	Requests	Execute
TimeSynchronization		x

DM-TS-B compliant devices must support the Local_Time and Local_Date properties of the Device object.

Device Management-UTCTimeSynchronization-B (DM-UTC-B)

The B device interprets time synchronization messages from the A device.

BACnet Service	Requests	Execute
UTCTimeSynchronization		x

DM-TS-B compliant devices must support the Local_Time, Local_Date, UTC_Offset, and Daylight_Savings_Status properties of the Device object.

Device Management-ReinitializeDevice-B (DM-RD-B)

The B device performs reinitialization requests from the A device. The optional password field shall be supported.

BACnet Service	Requests	Execute
ReinitializeDevice		x

Device Management-Backup and Restore-B (DM-BR-B)

The B device provides its configuration file to the A device and allows the A device to write this file to recover its configuration in the event of a failure.

BACnet Service	Requests	Execute
AtomicReadFile		x
AtomicWriteFile		x
ReinitializeDevice		x

DM-BR-B compliant devices must support the features of device B. Once a Restore procedure has been initiated on the device, the Read_Only property of configuration File objects shall contain the value FALSE and the File_Size property of the configuration File objects shall be writable if the size of the configuration file can change based on the device's configuration.

If the configuration file objects are not guaranteed to exist once a Restore procedure has been initiated, then the device must support execution of the CreateObject service.

Device Management-List Manipulation-B (DM-LM-B)

The B device responds to requests to add or remove list elements.

BACnet Service	Requests	Execute
AddListElement		x
RemoveListElement		x

Device Management-Object Creation and Deletion-B (DM-OCD-B)

The B device creates and deletes object instances based on requests from the A device. The object types whose dynamic creation and deletion is supported shall be enumerated in the Standard Object Types Supported section of the device B's PICS.

BACnet Service	Requests	Execute
CreateObject		x
DeleteObject		x

4.8.6 “Native” Operation of the B-BC

If the BACnet/IP Controller is switched on, not all objects that are supported by the system are present. In this still unconfigured state, the following objects are present for the connected modules:

- Analog Input Object
- Analog Output Object
- Binary Input Object
- Binary Output Object
- Calendar Object
- Device Object
- File Object
- Scheduler Object

The analog and digital Input and Output Objects are linked with the module data. A configuration or programming of the controller is not necessary for the creation and linking of these objects in native operation. The creation of the objects for the connected modules takes place automatically.

In native operation, the object name is formed from the object type, an underscore and the instance number. For example, the instance number 2 indicates that this is the second created object.

Example:

An analog output object with an instance number of 4 receives the object name “AnalogOutput_4”.



Note

Note the maximum of supported objects!

For the 750-831 BACnet/IP-Controller a maximum amount of 1000 objects can be created.

For the variant 750-831/000-002 a maximum amount of 256 objects can be created for the connected modules.



Information

Additional Information

In the BACnet Configurator, you have, among other things, the possibility of changing object names. These must be unique within a device.



Information

Additional Information

You can program objects that are not automatically created in the WAGO-I/O-PRO. For more information, please refer to the document “BACnet – Objects, Properties, Services” on the website <http://www.wago.com> in the “Documentation” area.

In the following, all objects that are automatically created in the native area are listed. Which of these objects are actually installed depends on the connected modules.

4.8.6.1 Analog Input Object

The Analog Input object defines a standardized object whose properties represent the externally visible characteristics of an analog input.

The Analog Input Object and its properties are summarized in the following table. The properties are described in section “List of all BACnet Properties in Native Operation”.

Table 33: Properties of the Analog Input Object

Property	Data Type	Default Value	Writable via BACnet by means of
Object_Identifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	Analog_Input_X	-
Object_Type	BACnetObjectType	ANALOG_INPUT	-
Present_Value	REAL	-	-
Description	CharacterString	*see footnote	WriteProperty
Device_Type	CharacterString	' '	WriteProperty
Status_Flags	BACnetStatusFlags	'000'	-
Event_State	BACnetEventState	State_Normal (0)	-
Reliability	BACnetReliability	0	WriteProperty
Out_Of_Service	BOOLEAN	'0'	-
Update_Interval	Unsigned	0	-
Units	BACnetEngineeringUnits	NO_UNITS (95)	WriteProperty
Min_Pres_Value	REAL	0	-
Max_Pres_Value	REAL	0	-
Resolution	REAL	0	-
COV_Increment	REAL	0	WriteProperty
Time_Delay	Unsigned	0	WriteProperty
Notification_Class	Unsigned	0	WriteProperty
High_Limit	REAL	0	WriteProperty
Low_Limit	REAL	0	WriteProperty
Deadband	REAL	0	WriteProperty
Limit_Enable	BACnetLimitEnable	'111'	WriteProperty
Event_Enable	BACnetEventTransitionBits	'111'	WriteProperty
Acked_Transitions	BACnetEventTransitionBits	'111'	-
Notify_Type	BACnetNotifyType	Alarm (0)	WriteProperty
Event_Time_Stamps	BACnetARRAY[3] for BACnet-TimeStamp	UNSPECIFIED	-
Message_Text (512)	BACnetARRAY[3] of CharacterString	' '	WriteProperty

* corresponds to the item number of the module

4.8.6.2 Analog Output Object

The Analog Output object defines a standardized object whose properties represent the externally visible characteristics of an analog output.

The Analog Output Object and its properties are summarized in the following table. The properties are described in section “List of all BACnet Properties in Native Operation”.

Table 34: Properties of the Analog Output Object

Property	Data Type	Default Value	Writable via BACnet by means of
Object_Identifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	Analog_Output_X	-
Object_Type	BACnetObjectType	ANALOG_OUTPUT	-
Present_Value	REAL	-	-
Description	CharacterString	*see footnote	WriteProperty
Device_Type	CharacterString	' '	WriteProperty
Status_Flags	BACnetStatusFlags	'000'	-
Event_State	BACnetEventState	State_Normal (0)	-
Reliability	BACnetReliability	0	WriteProperty
Out_Of_Service	BOOLEAN	'0'	-
Units	BACnetEngineeringUnits	NO_UNITS (95)	WriteProperty
Min_Pres_Value	REAL	0	-
Max_Pres_Value	REAL	0	-
Resolution	REAL	0	-
Priority_Array	BACnetPriorityArray	-	-
Relinquish_Default	REAL	0	WriteProperty
COV_Increment	REAL	0	WriteProperty
Time_Delay	Unsigned	0	WriteProperty
Notification_Class	Unsigned	0	WriteProperty
High_Limit	REAL	0	WriteProperty
Low_Limit	REAL	0	WriteProperty
Deadband	REAL	0	WriteProperty
Limit_Enable	BACnetLimitEnable	'111'	WriteProperty
Event_Enable	BACnetEventTransitionBits	'111'	WriteProperty
Acked_Transitions	BACnetEventTransitionBits	'111'	-
Notify_Type	BACnetNotifyType	Alarm (0)	WriteProperty
Event_Time_Stamps	BACnetARRAY[3] for BACnetTimeStamp	UNSPECIFIED	-
Message_Text (512)	BACnetARRAY[3] of CharacterString	' '	WriteProperty

*corresponds to the item number of the module

4.8.6.3 Binary Input Object

The Binary Input object defines a standardized object whose properties represent the externally visible characteristics of a binary input. A “binary input” is a physical device or hardware input that can be in only one of two distinct states. In this description, those states are referred to as ACTIVE and INACTIVE. A typical use of a binary input is to indicate whether a particular piece of mechanical equipment, such as a fan or pump, is running or idle. The state ACTIVE corresponds to the situation when the equipment is on or running, and INACTIVE corresponds to the situation when the equipment is off or idle.

In some applications, electronic circuits may reverse the relationship between the application-level logical states ACTIVE and INACTIVE and the physical state of the underlying hardware. For example, a normally open relay contact may result in an ACTIVE state when the relay is energized, while a normally closed relay contact may result in an INACTIVE state when the relay is energized. The Binary Input object provides for this possibility by including a Polarity property

The Binary Input Object and its properties are summarized in the following table. The properties are described in section “List of all BACnet Properties in Native Operation”.

Table 35: Properties of the Binary Input Object

Property	Data type	Default Value	Writable via BACnet by means of
Object_Identifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	Binary_Input_X	-
Object_Type	BACnetObjectType	BINARY_INPUT	-
Present_Value	BACnetBinaryPV	-	-
Description	CharacterString	*see footnote	WriteProperty
Device_Type	CharacterString	' '	WriteProperty
Status_Flags	BACnetStatusFlags	'000'	-
Event_State	BACnetEventState	State_Normal (0)	-
Reliability	BACnetReliability	0	WriteProperty
Out_Of_Service	BOOLEAN	'0'	-
Polarity	BACnetPolarity	0	WriteProperty
Inactive_Text	CharacterString	“inactive”	WriteProperty
Active_Text	CharacterString	“active”	WriteProperty
Change_Of_State_Time	BACnetDateTime	UNSPECIFIED	-
Change_Of_State_Count	Unsigned	0	-
Time_Of_State_Count_Reset	BACnetDateTime	UNSPECIFIED	-
Elapsed_Active_Time	Unsigned32	0	-
Time_Of_Active_Time_Reset	BACnetDateTime	UNSPECIFIED	-
Time_Delay	Unsigned	0	WriteProperty
Notification_Class	Unsigned	0	WriteProperty
Alarm_Value	BACnetBinaryPV	1	WriteProperty
Event_Enable	BACnetEventTransitionBits	'111'	WriteProperty
Acked_Transitions	BACnetEventTransitionBits	'111'	-
Notify_Type	BACnetNotifyType	Alarm (0)	WriteProperty
Event_Time_Stamps	BACnetARRAY[3] for BACnetTimeStamp	UNSPECIFIED	-

Table 35: Properties of the Binary Input Object

Property	Data type	Default Value	Writable via BACnet by means of
Message_Text (512)	BACnetARRAY[3] of CharacterString	' '	WriteProperty

*corresponds to the item number of the module

4.8.6.4 Binary Output Object

The Binary Output object defines a standardized object whose properties represent the externally visible characteristics of a binary output. A “binary out-put” is a physical device or hardware output that can be in only one of two distinct states. In this description, those states are referred to as ACTIVE and INACTIVE. A typical use for a Binary Output is the switching of a mechanical device, for example a ventilator or a pump. The state ACTIVE corresponds to the situation when the equipment is on or running, and INACTIVE corresponds to the situation when the equipment is off or idle.

In some applications, electronic circuits may reverse the relationship between the application-level logical states ACTIVE and INACTIVE and the physical state of the underlying hardware. For example, a normally open relay contact may result in an ACTIVE state (device energized) when the relay is energized, while a normally closed relay contact may result in an ACTIVE state (device energized) when the relay is not energized. The Binary Output Object provides for this possibility by including a Polarity property.

The Binary Output Object and its properties are summarized in the following table. The properties are described in section “List of all BACnet Properties in Native Operation”.

Table 36: Properties of the Binary Output Object

Property	Data type	Default Value	Writable via BACnet by means of
Object_Identifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	Binary_Output_X	-
Object_Type	BACnetObjectType	BINARY_OUTPUT	-
Present_Value	BACnetBinaryPV	-	-
Description	CharacterString	*see footnote	WriteProperty
Device_Type	CharacterString	' '	WriteProperty
Status_Flags	BACnetStatusFlags	'000'	-
Event_State	BACnetEventState	State_Normal (0)	-
Reliability	BACnetReliability	0	WriteProperty
Out_Of_Service	BOOLEAN	'0'	-
Polarity	BACnetPolarity	0	WriteProperty
Inactive_Text	CharacterString	“inactive”	WriteProperty
Active_Text	CharacterString	“active”	WriteProperty
Change_Of_State_Time	BACnetDateTime	UNSPECIFIED	-
Change_Of_State_Count	Unsigned	0	-
Time_Of_State_Count_Reset	BACnetDateTime	UNSPECIFIED	-

Table 36: Properties of the Binary Output Object

Property	Data type	Default Value	Writable via BACnet by means of
Elapsed_Active_Time	Unsigned32	0	-
Time_Of_Active_Time_Reset	BACnetDateTime	UNSPECIFIED	-
Minimum_Off_Time	Unsigned32	1	WriteProperty
Minimum_On_Time	Unsigned32	1	WriteProperty
Priority_Array	BACnetPriorityArray	-	-
Relinquish_Default	BACnetBinaryPV	0	WriteProperty
Time_Delay	Unsigned	0	WriteProperty
Notification_Class	Unsigned	0	WriteProperty
Feedback_Value	BACnetBinaryPV	0	-
Event_Enable	BACnetEventTransition Bits	'111'	WriteProperty
Acked_Transitions	BACnetEventTransition Bits	'111'	-
Notify_Type	BACnetNotifyType	Alarm (0)	WriteProperty
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	UNSPECIFIED	-
Message_Text (512)	BACnetARRAY[3] of CharacterString	' '	WriteProperty

*corresponds to the item number of the module

4.8.6.5 Calendar Object

The Calendar object defines a standardized object used to describe a list of calendar dates, which might be thought of as “holidays,” “special events,” or simply as a list of dates.

The Calendar Object and its properties are summarized in the following table. The properties are described in section “List of all BACnet Properties in Native Operation”.

Table 37: Properties of the Calendar Object

Property	Data type	Default Value	Writable via BACnet by means of
Object_Identifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	“CalendarX”	-
Object_Type	BACnetObjectType	CALENDAR	-
Description	CharacterString	“”	WriteProperty
Present_Value	BOOLEAN	-	WriteProperty
Date_List	List of BACnetCalendarEntry	-	WriteProperty
Message_Text (512)	BACnetARRAY[3] of CharacterString	' '	WriteProperty

4.8.6.6 Device Object

The Device object defines a standardized object whose properties represent the externally visible characteristics of a BACnet Device. There shall be exactly one Device object in each BACnet Device. A Device object is referenced by its Object_Identifier property, which is not only unique to the BACnet Device that maintains this object but is also unique throughout the BACnet internetwork.

The Device Object and its properties are summarized in the following table. The properties are described in section “List of all BACnet Properties in Native Operation”.

Table 38: Properties of the Device Object

Property	Data type	Default Value	Writable via BACnet by means of
Object_Identifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	“Unnamed”	-
Object_Type	BACnetObjectType	DEVICE	-
System_Status	BACnetDeviceStatus	OPERATIONAL (0)	-
Vendor_Name	CharacterString	“WAGO Kontakt-technik GmbH & Co. KG”	-
Vendor_Identifier	Unsigned16	222	-
Model_Name	CharacterString	750-831	-
Firmware_Revision	CharacterString	-	-
Application_Software_Version	CharacterString	-	-
Location	CharacterString	“”	WriteProperty
Description	CharacterString	“”	WriteProperty
Protocol_Version	Unsigned	-	-
Protocol_Revision	Unsigned	-	-
Protocol_Services_Supported	BACnetServices-Supported	-	-
Protocol_Object_Types_Supported	BACnetObjectTypes-Supported	-	-
Object_List	BACnetARRAY[N] des BACnetObjectIdentifier	-	-
Max_APDU_Length_Accepted	Unsigned	480	-
Segmentation_Supported	BACnetSegmentation	SEGMENTED_BOTH (0)	-
Max_Segments_Accepted	Unsigned	16	-
Local_Time	Time	-	-
Local_Date	Date	-	-
UTC_Offset	INTEGER	0	WriteProperty
Daylight_Savings_Status	BOOLEAN	'0'	WriteProperty
APDU_Segment_Timeout	Unsigned	2000	-
APDU_Timeout	Unsigned	3000	-
Number_Of_APDU_Retries	Unsigned	3	-
Device_Address_Binding	List of BACnetAddressBinding	-	-

Table 38: Properties of the Device Object

Property	Data type	Default Value	Writable via BACnet by means of
Database_Revision	Unsigned	-	-
Configuration_Files	BACnetARRAY[N] des BACnetObjectIdentifier	-	-
Last_Restore_Time	BACnetTimeStamp	UNSPECIFIED	-
Backup_Failure_Timeout	Unsigned16	10	-
Active_COV_Subscriptions	List of BACnetCOV-Subscription	-	-
Message_Text (512)	BACnetARRAY[3] of CharacterString	' '	WriteProperty

4.8.6.7 File Object

A File Object is created for each BACnet-relevant file in the file system. The File Object is described in the Standard and defines file properties that are accessed by file access services such as the AtomicReadFile Service.

The File Object and its properties are summarized in the following table. The properties are described in section “List of all BACnet Properties in Native Operation”.

Table 39: Properties of the File Object

Properties	Data type	Default Value	Writable via BACnet by means of
Object_Identifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	“FileX”	-
Object_Type	BACnetObjectType	FILE	-
Description	CharacterString	“”	WriteProperty
File_Type	CharacterString	“”	WriteProperty
File_Size	Unsigned	-	WriteProperty
Modification_Date	BACnetDateTime	UNSPECIFIED	-
Archive	BOOLEAN	0	WriteProperty
Read_Only	BOOLEAN	TRUE	WriteProperty
File_Access_Method	BACnetFileAccessMethod	STREAM_ACCESS	-
Message_Text (512)	BACnetARRAY[3] of CharacterString	' '	WriteProperty

4.8.6.8 Schedule Object

The Schedule object defines a standardized object used to describe a periodic schedule that may recur during a range of dates, with optional exceptions at arbitrary times on arbitrary dates. The Schedule Object also serves as a binding between these scheduled times and the writing of specified “values” to specific properties of specific objects at those times.

Schedules are divided into days, of which there are two types: normal days within a week and exception days. Both types of days can specify scheduling events for

either the full day or portions of a day. A priority system determines which planned event is in control at any given time.

The current state of the Schedule object is represented by the value of its Present_Value property, which is normally calculated using the time/value pairs from the Weekly_Schedule and Exception_Schedule properties. A default value is used when no schedules are in effect. Details of this calculation are provided in the description of the Present_Value property.

Versions of the Schedule Object prior to Protocol_Revision 4 only support schedules that define an entire day (from midnight to midnight). For compatibility with these versions, this whole day behavior can be achieved by using a specific schedule format. Weekly_Schedule and Exception_Schedule values that begin at 00:00, and do not use any NULL values, will define schedules for the entire day. Property values in this format will produce the same results in all versions of the Schedule Object.

The Schedule Object and its properties are summarized in the following table. The properties are described in section “List of all BACnet Properties in Native Operation”.

Table 40: Properties of the Schedule Object

Property	Data Type	Default Value	Writable via BACnet by means of
Object_Identifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	“ScheduleX”	-
Object_Type	BACnetObjectType	SCHEDULE	-
Present_Value	Any	-	-
Description	CharacterString	“”	WriteProperty
Effective_Period	BACnetDateRange	UNSPECIFIED	WriteProperty
Weekly_Schedule	BACnetARRAY[7]of BACnetDailySchedule	-	WriteProperty
Exception_Schedule	BACnetARRAY[N]of BACnetSpecialEvent	-	WriteProperty
Schedule_Default	Any	-	WriteProperty
List_Of_Object_Property_References	List of BACnetDevice	-	WriteProperty
Priority_For_Writing	ObjectPropertyReference	16	WriteProperty
Status_Flags	Unsigned(1..16)	‘0000’	-
Reliability	BACnetStatusFlags	-	-
Out_Of_Service	BACnetReliability	-	WriteProperty
Message_Text (512)	BACnetARRAY[3] of CharacterString	‘ ’	WriteProperty

5 Mounting

5.1 Installation Position

Along with horizontal and vertical installation, all other installation positions are allowed.



Note

Use an end stop in the case of vertical mounting!

In the case of vertical assembly, an end stop has to be mounted as an additional safeguard against slipping.

WAGO order no. 249-116 End stop for DIN 35 rail, 6 mm wide

WAGO order no. 249-117 End stop for DIN 35 rail, 10 mm wide

5.2 Overall Configuration

The maximum total length of a fieldbus node without fieldbus coupler/controller is 780 mm including end module. The width of the end module is 12 mm. When assembled, the I/O modules have a maximum length of 768 mm.

Examples:

- 64 I/O modules with a 12 mm width can be connected to a fieldbus coupler/controller.
- 32 I/O modules with a 24 mm width can be connected to a fieldbus coupler/controller.

Exception:

The number of connected I/O modules also depends on the type of fieldbus coupler/controller is used. For example, the maximum number of stackable I/O modules on one PROFIBUS DP/V1 fieldbus coupler/controller is 63 with no passive I/O modules and end module.

NOTICE

Observe maximum total length of a fieldbus node!

The maximum total length of a fieldbus node without fieldbus coupler/controller and without using a 750-628 I/O Module (coupler module for internal data bus extension) may not exceed 780 mm.

Also note the limitations of individual fieldbus couplers/controllers.



Note

Increase the total length using a coupler module for internal data bus extension!

You can increase the total length of a fieldbus node by using a 750-628 I/O Module (coupler module for internal data bus extension). For such a configuration, attach a 750-627 I/O Module (end module for internal data bus extension) after the last I/O module of a module assembly. Use an RJ-45 patch cable to connect the I/O module to the coupler module for internal data bus extension of another module block.

This allows you to segment a fieldbus node into a maximum of 11 blocks with maximum of 10 I/O modules for internal data bus extension.

The maximum cable length between two blocks is five meters.

More information is available in the manuals for the 750-627 and 750-628 I/O Modules.

5.3 Mounting onto Carrier Rail

5.3.1 Carrier Rail Properties

All system components can be snapped directly onto a carrier rail in accordance with the European standard EN 50022 (DIN 35).

NOTICE

Do not use any third-party carrier rails without approval by WAGO!

WAGO Kontakttechnik GmbH & Co. KG supplies standardized carrier rails that are optimal for use with the I/O system. If other carrier rails are used, then a technical inspection and approval of the rail by WAGO Kontakttechnik GmbH & Co. KG should take place.

Carrier rails have different mechanical and electrical properties. For the optimal system setup on a carrier rail, certain guidelines must be observed:

- The material must be non-corrosive.
- Most components have a contact to the carrier rail to ground electro-magnetic disturbances. In order to avoid corrosion, this tin-plated carrier rail contact must not form a galvanic cell with the material of the carrier rail which generates a differential voltage above 0.5 V (saline solution of 0.3 % at 20°C).
- The carrier rail must optimally support the EMC measures integrated into the system and the shielding of the I/O module connections.
- A sufficiently stable carrier rail should be selected and, if necessary, several mounting points (every 20 cm) should be used in order to prevent bending and twisting (torsion).
- The geometry of the carrier rail must not be altered in order to secure the safe hold of the components. In particular, when shortening or mounting the carrier rail, it must not be crushed or bent.
- The base of the I/O components extends into the profile of the carrier rail. For carrier rails with a height of 7.5 mm, mounting points are to be riveted under the node in the carrier rail (slotted head captive screws or blind rivets).
- The metal springs on the bottom of the housing must have low-impedance contact with the DIN rail (wide contact surface is possible).

5.3.2 WAGO DIN Rails

WAGO carrier rails meet the electrical and mechanical requirements shown in the table below.

Table 41: WAGO DIN Rails

Item No.	Description
210-112	35 × 7.5; 1 mm; steel; bluish, tinned, chromed; slotted
210-113	35 × 7.5; 1 mm; steel; bluish, tinned, chromed; unslotted
210-197	35 × 15; 1.5 mm; steel; bluish, tinned, chromed; slotted
210-114	35 × 15; 1.5 mm; steel; bluish, tinned, chromed; unslotted
210-118	35 × 15; 2.3 mm; steel; bluish, tinned, chromed; unslotted
210-198	35 × 15; 2.3 mm; copper; unslotted
210-196	35 × 8.2; 1.6 mm; aluminum; unslotted

5.4 Spacing

The spacing between adjacent components, cable conduits, casing and frame sides must be maintained for the complete fieldbus node.

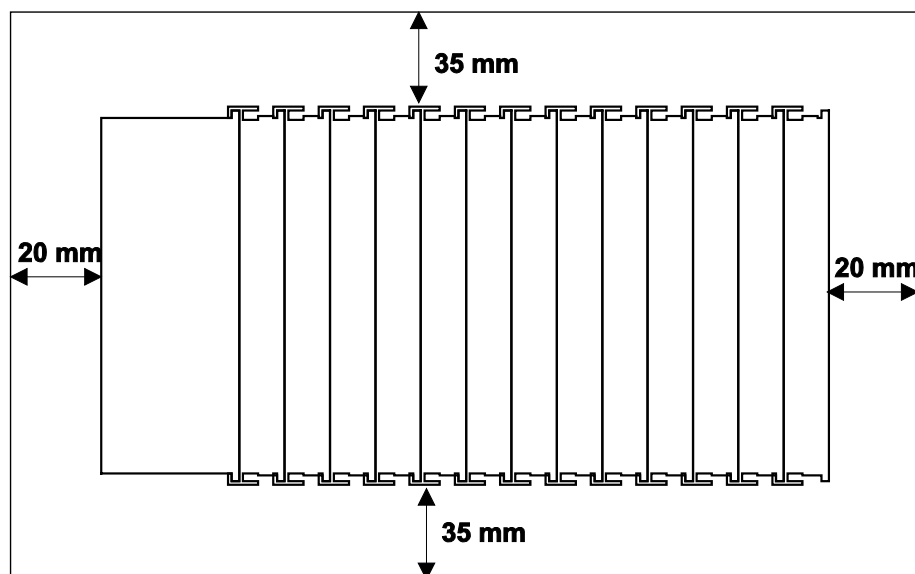


Figure 30: Spacing

The spacing creates room for heat transfer, installation or wiring. The spacing to cable conduits also prevents conducted electromagnetic interferences from influencing the operation.

5.5 Mounting Sequence

Fieldbus couplers/controllers and I/O modules of the WAGO-I/O-SYSTEM 750 are snapped directly on a carrier rail in accordance with the European standard EN 50022 (DIN 35).

The reliable positioning and connection is made using a tongue and groove system. Due to the automatic locking, the individual devices are securely seated on the rail after installation.

Starting with the fieldbus coupler/controller, the I/O modules are mounted adjacent to each other according to the project design. Errors in the design of the node in terms of the potential groups (connection via the power contacts) are recognized, as the I/O modules with power contacts (blade contacts) cannot be linked to I/O modules with fewer power contacts.

CAUTION

Risk of injury due to sharp-edged blade contacts!

The blade contacts are sharp-edged. Handle the I/O module carefully to prevent injury.

NOTICE

Insert I/O modules only from the proper direction!

All I/O modules feature grooves for power jumper contacts on the right side. For some I/O modules, the grooves are closed on the top. Therefore, I/O modules featuring a power jumper contact on the left side cannot be snapped from the top. This mechanical coding helps to avoid configuration errors, which may destroy the I/O modules. Therefore, insert I/O modules only from the right and from the top.

Note



Don't forget the bus end module!

Always plug a bus end module (750-600) onto the end of the fieldbus node! You must always use a bus end module at all fieldbus nodes with WAGO-I/O-SYSTEM 750 fieldbus couplers/controllers to guarantee proper data transfer.

5.6 Inserting and Removing Devices

NOTICE

Perform work on devices only if they are de-energized!

Working on energized devices can damage them. Therefore, turn off the power supply before working on the devices.

5.6.1 Inserting the Fieldbus Coupler/Controller

1. When replacing the fieldbus coupler/controller for an already available fieldbus coupler/controller, position the new fieldbus coupler/controller so that the tongue and groove joints to the subsequent I/O module are engaged.
2. Snap the fieldbus coupler/controller onto the carrier rail.
3. Use a screwdriver blade to turn the locking disc until the nose of the locking disc engages behind the carrier rail (see the following figure). This prevents the fieldbus coupler/controller from canting on the carrier rail.

With the fieldbus coupler/controller snapped in place, the electrical connections for the data contacts and power contacts (if any) to the possible subsequent I/O module are established.

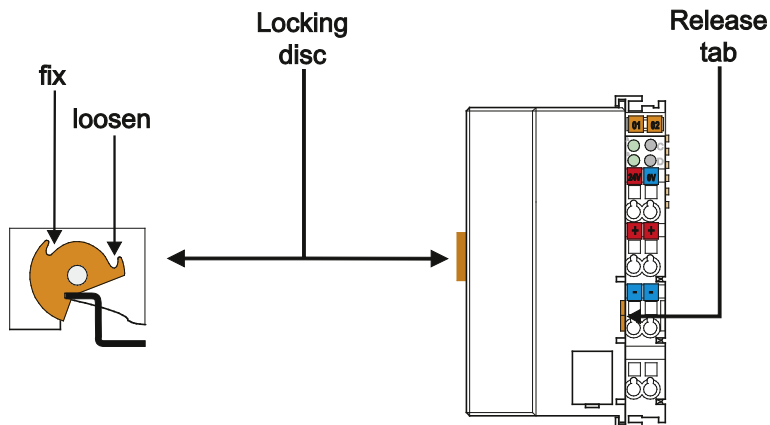


Figure 31: Release Tab of Extended ECO Fieldbus Coupler (Example)

5.6.2 Removing the Fieldbus Coupler/Controller

1. Use a screwdriver blade to turn the locking disc until the nose of the locking disc no longer engages behind the carrier rail.
2. Remove the fieldbus coupler/controller from the assembly by pulling the release tab.

Electrical connections for data or power contacts to adjacent I/O modules are disconnected when removing the fieldbus coupler/controller.

5.6.3 Inserting the I/O Module

1. Position the I/O module so that the tongue and groove joints to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are engaged.



Figure 32: Insert I/O Module (Example)

2. Press the I/O module into the assembly until the I/O module snaps into the carrier rail.

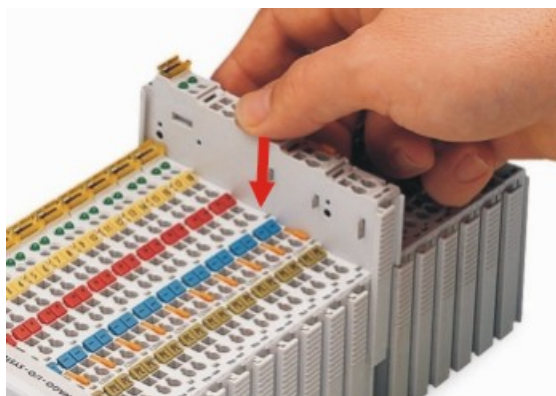


Figure 33: Snap the I/O Module into Place (Example)

With the I/O module snapped in place, the electrical connections for the data contacts and power jumper contacts (if any) to the fieldbus coupler/controller or to the previous or possibly subsequent I/O module are established.

5.6.4 Removing the I/O Module

1. Remove the I/O module from the assembly by pulling the release tab.

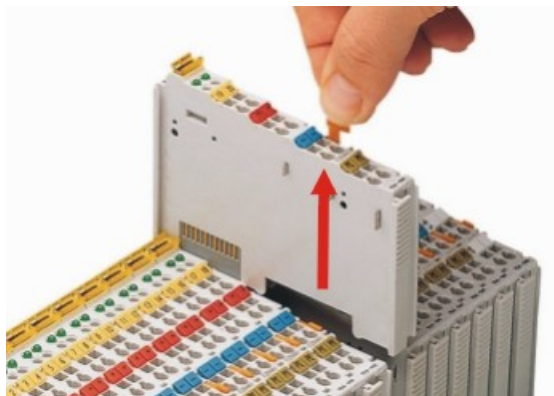


Figure 34: Removing the I/O Module (Example)

Electrical connections for data or power jumper contacts are disconnected when removing the I/O module.

6 Connect Devices

6.1 Data Contacts/Internal Bus

Communication between the fieldbus coupler/controller and the I/O modules as well as the system supply of the I/O modules is carried out via the internal bus. It is comprised of 6 data contacts, which are available as self-cleaning gold spring contacts.

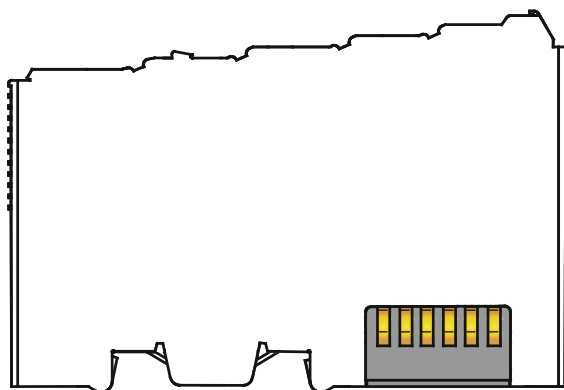


Figure 35: Data Contacts

NOTICE

Do not place the I/O modules on the gold spring contacts!

Do not place the I/O modules on the gold spring contacts in order to avoid soiling or scratching!

NOTICE



Ensure that the environment is well grounded!

The devices are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the devices, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, e.g. data contacts.

6.2 Power Contacts/Field Supply

⚠ CAUTION

Risk of injury due to sharp-edged blade contacts!

The blade contacts are sharp-edged. Handle the I/O module carefully to prevent injury.

Self-cleaning power jumper contacts used to supply the field side are located on the right side of most of the fieldbus couplers/controllers and on some of the I/O modules. These contacts come as touch-proof spring contacts. As fitting counterparts the I/O modules have male contacts on the left side.

Power jumper contacts

Blade	0	0	3	2
Spring	0	3	3	2

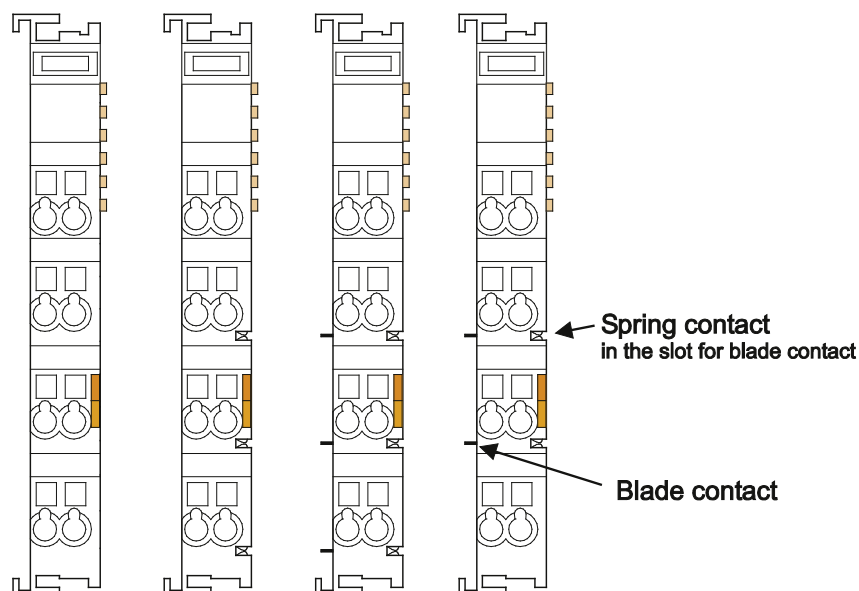


Figure 36: Example for the Arrangement of Power Contacts

Note



Field bus node configuration and test via smartDESIGNER

With the WAGO ProServe® Software smartDESIGNER, you can configure the structure of a fieldbus node. You can test the configuration via the integrated accuracy check.

6.3 Connecting a Conductor to the CAGE CLAMP®

The WAGO CAGE CLAMP® connection is appropriate for solid, stranded and finely stranded conductors.



Note

Only connect one conductor to each CAGE CLAMP®!

Only one conductor may be connected to each CAGE CLAMP®.

Do not connect more than one conductor at one single connection!

If more than one conductor must be routed to one connection, these must be connected in an up-circuit wiring assembly, for example using WAGO feed-through terminals.

1. For opening the CAGE CLAMP® insert the actuating tool into the opening above the connection.
2. Insert the conductor into the corresponding connection opening.
3. For closing the CAGE CLAMP® simply remove the tool. The conductor is now clamped firmly in place.

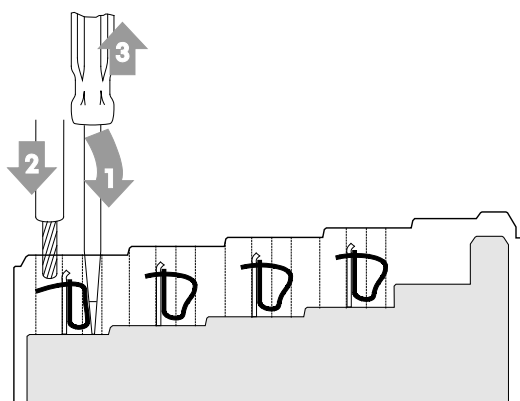


Figure 37: Connecting a Conductor to a CAGE CLAMP®

7 Function Description

7.1 Operating System

7.1.1 Run-up



Note

The mode selector switch may not be located in the lower position!

The mode selector switch may not be set at the bottom position during run-up!

The fieldbus controller begins running up after switching on the power supply or after a reset. The internal PFC program is then transferred to the RAM.

During the initialization phase, the fieldbus controller detects the I/O modules and the current configuration and sets the variables to 0 or FALSE, or to an initial value specified by the PFC program. The flags retain their status. During this phase the I/O LED will flash red.

When run-up is successful, the I/O LED then stays lit continuously in green.



Information

More information about the LED Signaling

Read the detailed description for the evaluation of the displayed LED state in the section "Diagnostics" > ... > "LED Signaling".



Information

Additional Information:

BACnet-specific information about the starting behavior is available in chapter "Start-Up" → ... → "Information about the Starting Behavior regarding BACnet and WAGO-I/O-PRO Program".

7.1.2 PFC Cycle

After error-free run-up, the PFC cycle starts with the mode selector switch at the top position, or on a Start command from WAGO-I/O-PRO. The input and output data for the fieldbus, I/O modules and the timer values are read. The PFC program contained in the RAM is then processed, after which the output data for the fieldbus and I/O modules is written to the process image. At the end of the PFC cycle, the operating system functions are executed for diagnostics and communication (among other things) and the timer values are updated. The new cycle begins by reading in of the input and output data and the timer values.

The operating mode is changed ("STOP"/"RUN") at the end of a PFC cycle.

The cycle time is the time from the beginning of the PFC program up to the next beginning of the cycle. If a loop is programmed within the PFC program, the PFC runtime and the PFC cycle time will be extended accordingly.

The inputs, outputs and timer values are not updated while the PFC program is being processed. Updating is performed only as defined at the end of the PFC program. As a result, it is not possible to wait on an event from the process or a set period to expire while a loop is in progress.

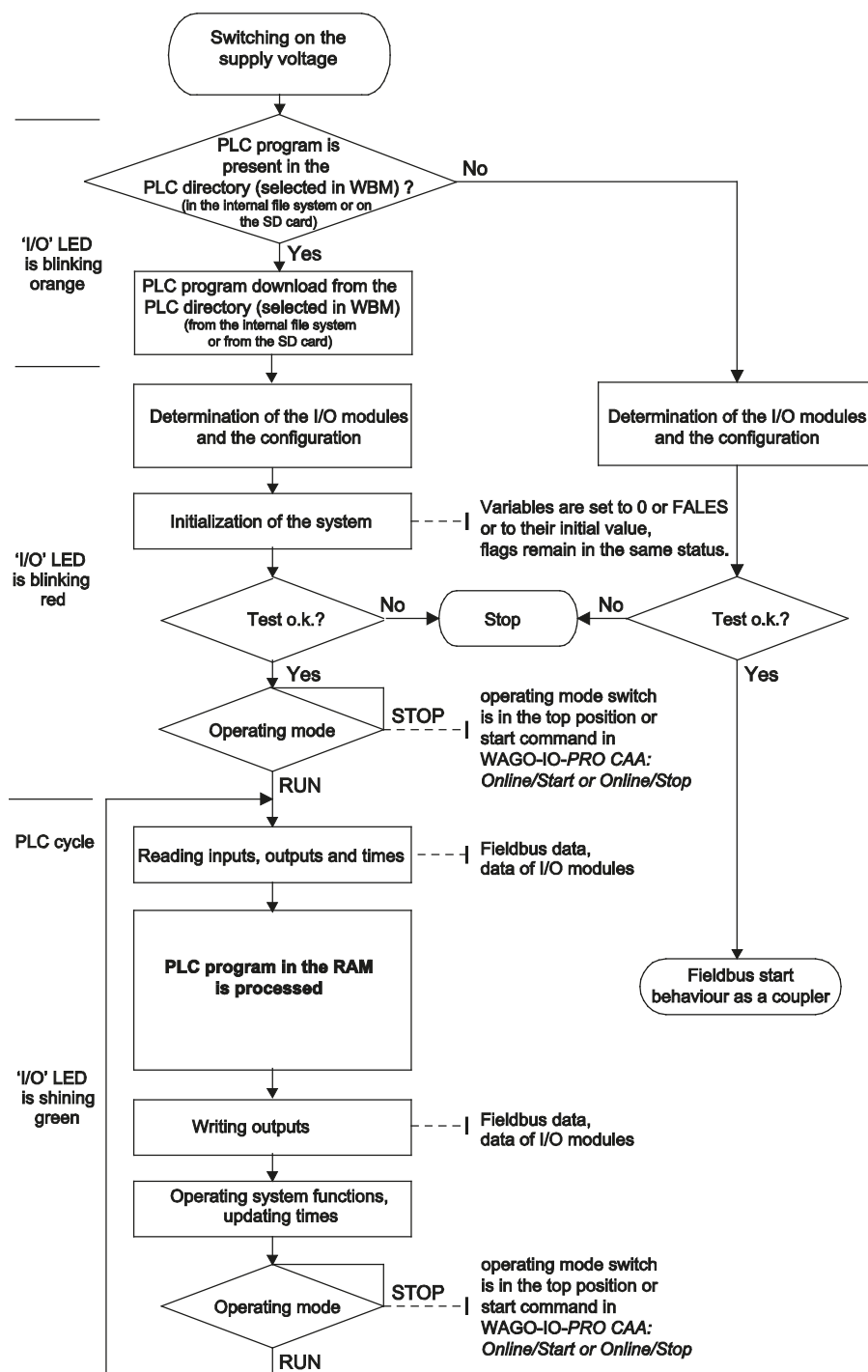


Figure 38: Run-up of the Controller

7.2 Process Data Architecture

The following sections provide a glimpse of the internal functioning, data processing and addressing in MODBUS communication.

BACnet process data, on the other hand, are not stored in a fixed, internal process image. Using the connected modules, the BACnet/IP Controller creates BACnet objects that represent the process data and that are not located in any directly addressable or visible process image.

7.2.1 Basic Structure

After switching on, the fieldbus controller identifies all I/O modules connected with the node that send or receive data (data width/bit width > 0).

In a node can consist of a mixed arrangement of analog input/analog output modules and digital input/digital output modules and special modules.

Note



Up to 99 I/O modules can be connected with the data bus extension modules. Using the bus extension coupler module 750-628 and end module for internal data bus extension 750-627 makes it possible to connect up to 99 modules to the BACnet/IP Controller.

Information



Additional Information

For the number of input and output bits or bytes of the individual I/O modules, refer to the corresponding description of the I/O modules.

The controller creates an internal local process image on the basis of the data width, the type of I/O module and the position of the module in the node. This process image is broken down into an input and an output data range.

The data of the digital I/O modules is bit-oriented; i.e., digital data is sent bit by bit. Analog I/O modules represent the group of byte-oriented modules – data is sent byte by byte.

Counter modules, DALI, MP bus, EnOcean and communication modules, for example, are included in this group of modules.

For both, the local input and output process image, the I/O module data is stored in the corresponding process image depending on the order in which the modules are connected to the fieldbus controller.

Note



Hardware changes can result in changes of the process image!

If the hardware configuration is changed by adding, changing, removing or reparametrisation of I/O modules with a data width > 0 bit, this result in a new process image structure. The process data addresses would then change. If adding I/O modules, the process data of all previous I/O modules has to be taken into account.

A memory range of 256 words (word 0...255) is initially available in the controller for the process image of the physical input and output data.

For the image of the MODBUS/PFC variables, the memory range of words 256...511 is reserved; meaning the image for the MODBUS/PFC variables is created behind the process image for the I/O module data.

If the quantity of module data is greater than 256 words, all the physical input and output data above this value is added to the end of the current process image in a memory range; i.e., attached behind the MODBUS/PFC variables (word 512...1275).

The range from word 1276 to word 1531 is not available to the user.

The subsequent range, starting from word 1532, is reserved for future protocol expansion and other PFC variables.

Access by the PLC to process data is made independently from the fieldbus system in all WAGO fieldbus controllers; access is always conducted through an application-related IEC-61131-3 program.
How the data is accessed from the fieldbus side depends on the fieldbus however.

A MODBUS/TCP Master can access the data for the BACnet/IP Controller via implemented MODBUS functions.

Information



Additional Information

For a detailed description of these fieldbus-specific data access methods, refer to the Section "MODBUS Functions."

7.2.2 Example of an Input Process Image

The following figure is an example of an input process image.
The configuration comprises 16 digital and 8 analog inputs.
The input process image thus has a data length of 8 words for the analog modules and 1 word for the digital modules; i.e., 9 words in total.

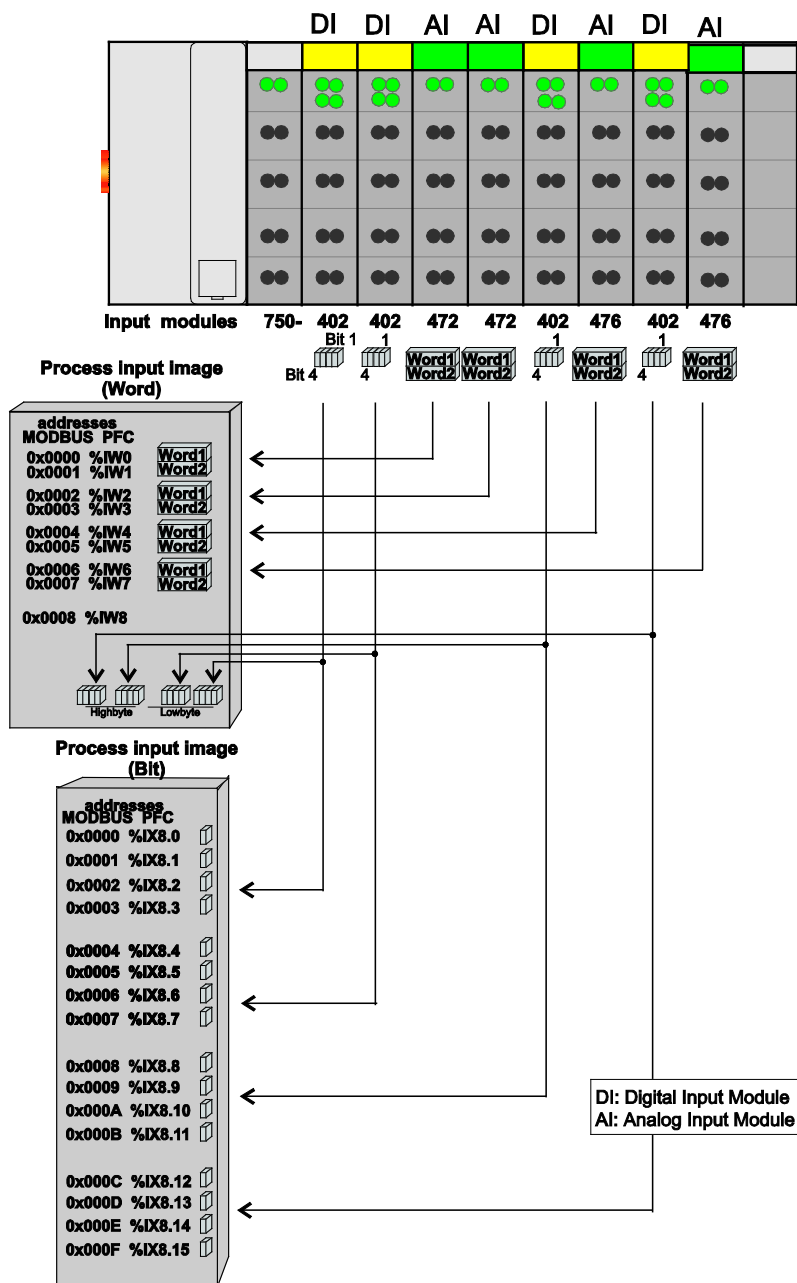


Figure 39: Example of Process Image for Input Data

7.2.3 Example of an Output Data Process Image

The following example for the output process image comprises 2 digital and 4 analog outputs. It comprises 4 words for the analog outputs and 1 word for the digital outputs; i.e., 5 words in total.

In addition, the output data can also be read back with an offset of 200_{hex} (0x0200) added to the MODBUS address.



Note

Data > 256 words can be read back by using the cumulative offset!

All output data greater than 256 words and, therefore located in the memory range 6000_{hex} (0x6000) to 66F9_{hex} (0x66F9) can be read back with an offset of 1000_{hex} (0x1000) added to the MODBUS address.

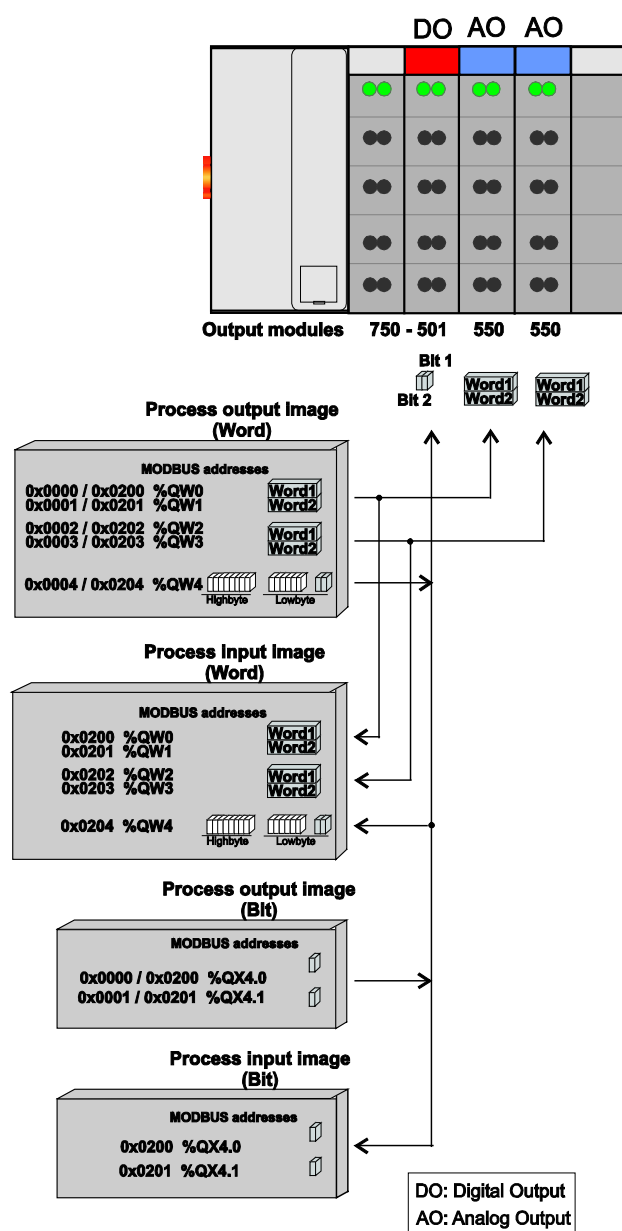


Figure 40: Example of Process Image for Output Data

7.2.4 MODBUS Process Data

For some I/O modules (and their variations), the structure of the process data depends on the fieldbus.

When applying the MODBUS protocol, the process image has a word structure (with word alignment). The internal mapping method for data greater than one byte conforms to the Intel format. The modules can be mapped directly via addresses with MODBUS.

Information



Additional Information

For the fieldbus-specific structure of the process values of any I/O module within the 750 or 753 Series of the WAGO-I/O-SYSTEM, refer to Section "Structure of process data for MODBUS/TCP".

7.3 Data Exchange

Exchange of process data takes place with BACnet/IP Controller using the BACnet/IP protocol or the MODBUS protocol.

The BACnet/IP Controller works according to the client server principle. The client requests services from the server. It subscribes, for example, to changes in value or sets limits for alarm/event reports. With its objects, the server maps and executes the service requests of the client.

A controller can set up a defined number of simultaneous connections (socket connections) to other network subscribers:

- 3 connections for HTTP (to read HTML pages from the fieldbus controller)
- 30 connections via MODBUS/TCP (to read or write input and output data of the fieldbus controller)
- 5 connections (when using the Ethernet.lib) via PFC (available in the PLC function for IEC 61131-3 application programs)
- 2 connections for WAGO-I/O-PRO (these connections are reserved for debugging the application program via ETHERNET. WAGO-I/O-PRO needs 2 connections at the same time for the debugging. However, only **one** programming tool can have access to the fieldbus controller)
- 10 connections for FTP
- 2 connections for SNMP

The maximum number of simultaneous connections can not be exceeded. Existing connections must first be terminated before new ones can be set up. The BACnet/IP Controller is essentially equipped with three interfaces for data exchange:

- the interface to the fieldbus (Master),
- the PLC function of the PFC (CPU) and
- the interface to the I/O modules.

Data exchange takes place between

- the fieldbus master and the I/O modules,
- the PLC function of the PFC (CPU) and the I/O modules
- between the fieldbus master and the PLC function of the PFC (CPU).

If MODBUS is used as the fieldbus, the MODBUS master accesses the data using the MODBUS functions implemented in the controller.

Data access is carried out with the aid of an IEC-61131-3 application program. Data addressing varies greatly here.

7.3.1 MODBUS Memory Areas

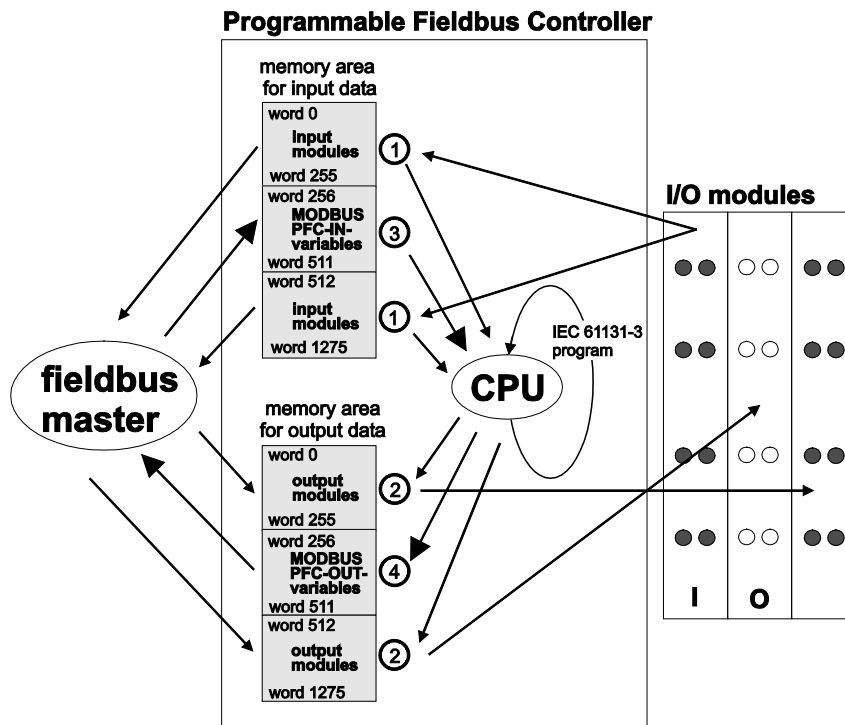


Figure 41: Memory Areas and Data Exchange

The controller process image contains the physical data for the bus modules. These have a value of 0 ... 255 and word 512 ... 1275.

- ① The input module data can be read by the CPU and by the fieldbus side.
- ② Likewise, data can be written to the output modules from the CPU and the fieldbus side.

The MODBUS PFC variables are stored in each of the memory areas for word 256 ... 511 between these sides.

- ③ The MODBUS-PFC input variables are written to the input memory area from the fieldbus side and read in by the CPU for processing.
- ④ The variables processed by the CPU using the IEC-61131-3 program are places in the output memory area, where they can be read out by the master.

The memory area for word 1276 ... 1531 is adjacent to the physical I/O module data. This area is reserved and may not be used.

The subsequent memory area, starting from word 1532, is reserved for future protocol expansion and other PFC variables.

In addition, all output data is mirrored in the BACnet/IP Controller to a memory area with the address offset 0x0200 and 0x1000. This allows output values to be read back in by adding 0x0200 or 0x1000 to the MODBUS address.

Other memory areas are also provided in the fieldbus controller, some of which cannot be accessed by the fieldbus side, however:

- **Data memory (1024 kByte)**
The data memory is a volatile RAM memory for creating variables that are not required for communication with the interfaces, but rather for internal processing procedures, such as calculation of results.
- **Program memory (1024 kByte)**
The IEC-61131-3 program is stored in the program memory. The code memory is a Flash ROM. When power is switched on, the program is transferred from the flash to the RAM memory. After error-free run-up, the PFC cycle starts with the mode selector switch at the top position, or on the Start command from the WAGO-I/O-PRO.
- **NOVRAM Remanent memory (28 kByte)**
The remanent memory is a non-volatile memory; i.e., all values of flags and variables, that are explicitly defined by "VAR RETAIN", are retained even after a loss of power. Memory management is performed automatically. The 28 kByte memory area is normally divided into an 16 kByte addressable range for flags (%MW0 ... %MW8191) and a 12 kByte retain area for variables without memory area addressing, that are defined by "VAR RETAIN".

Note



Markers are only remanent under "VAR RETAIN"!

Please note that the bit memory is only retentive if you have declared it as such under "VAR RETAIN".

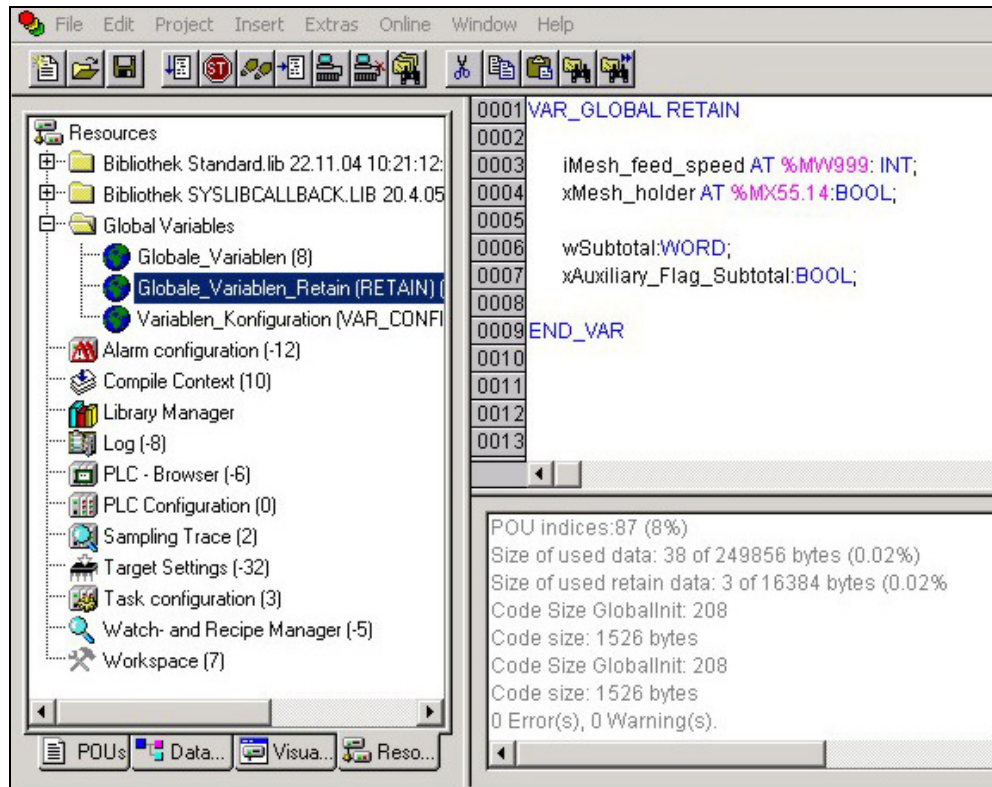


Figure 42: Example Declaration of Remanent Flags by "VAR RETAIN"

This breakdown of the NOVDRAM can be varied (see following explanation).

Note



NOVDRAM memory allocation can be changed in WAGO-I/O-PRO!

The breakdown of the NOVDRAM can be modified when required in the programming software WAGO-I/O-PRO/Register "Resources"/Dialog window "Target system settings".

The start address for the flag area is fixed. The area sizes and the start address for the retain memory can be varied.

We do recommend keeping the standard settings, however, in order to avoid any overlapping of the areas.

In these default settings the size of the flag area is set at 16#4000, followed by the retain memory, with the size 16#3000.

7.3.2 Addressing

Module inputs and outputs in a fieldbus controller are addressed internally as soon as they are started. The order in which the connected modules are addressed depends on the type of module that is connected (input module, output module). The process image is formed from these addresses. The physical arrangement of the I/O modules in the fieldbus node is arbitrary.

Note

**Use various options for addressing the bus terminals!**

Connected modules in more detail. It is essential that you understand these correlations in order to conduct conventional addressing by counting. The **WAGO I/O Configurator** is also available as a further addressing option. The Configurator can assist you in addressing and protocol assignment for the connected modules. You must select the connected modules in the I/O Configurator; the software then takes care of correct addressing (see following Figure).

The I/O Configurator is started from the *WAGO-I/O-PRO*.

For more details, refer to section “Programming the PFC Using *WAGO-I/O-PRO*” > ... > “Configuring the Fieldbus Controller using the I/O Configurator”.

7.3.2.1 Addressing of I/O Modules

Addressing first references complex modules (modules that occupy several bytes) in accordance with their physical order downstream of the fieldbus coupler/controller; i.e., they occupy addresses starting from word 0.

Following these is the data for the remaining modules, compiled in bytes (modules that occupy less than one byte). In this process, byte by byte is filled with this data in the physical order. As soon a complete byte is occupied by the bit oriented modules, the process begins automatically with the next byte.

Note

**Hardware changes can result in changes of the process image!**

If the hardware configuration is changed and/or expanded; this may result in a new process image structure. In this case, the process data addresses also change. If adding modules, the process data of all previous modules has to be taken into account.

Note

**Observe process data quantity!**

For the number of input and output bits or bytes of the individual IO modules please refer to the corresponding description of the IO modules.

Table 42: Data Width for I/O Modules

Data width > 1 byte (channel)	Data width = 1 bit (channel)
Analog input modules	Digital input modules
Analog output modules	Digital output modules
Input modules for thermocouples	Digital output modules with diagnostics (2 bits/channel)
Input modules for resistor sensors	Supply modules with fuse carrier/diagnostics
Pulse width output modules	Solid-state load relays
Interface modules	Relay output modules
Up/down counters	
I/O modules for angle and distance measurement	

7.3.2.2 Example of Addressing

A controller to which two digital input modules (2 DI), two digital output modules (2 DO) and two analog input modules (2 AI) and two analog output modules (2AO) are connected. The final element is an end module that is not taken into account for addressing.

Table 43: Example of Addressing

Counting order	Module	Function	Data width	Hardware Address
1.	750-467	2 AI / 0-10 Volt	2 x 16 bit	%IW0 u. %IW1
2.	750-400	2 DI	2 x 1 bit	%IX2.0 u. %IX2.1
3.	750-550	2 AO / 0-10 Volt	2 x 16 bit	%QW0 u. %QW1
4.	750-501	2 DO	2 x 1 bit	%QX2.0 u. %QX2.1
-	750-600	End module	none	-

Refer to the technical data for the specific modules for the data width. The analog input modules (AI) are mapped first in the process image. Analog modules are processed word-by-word (W).

Module 467 occupies 2 words here (1 word = 16 Bit); i.e.; the first word **%IW0** and the second word **%IW1** in the memory image. Note here that counting begins at "0".

The digital inputs (DI) are taken into account after this. These occupy 2 bits. Two complete words have been previously counted (Word 0 and 1). Now, counting is continued from Word 2 and 2 bits are added (Bit 0 and Bit 1). Words and bits are each separated by a decimal point. The hardware addresses are **%IX2.0** and **%IX2.1**.

The two analog output modules 750-550 (AO) are then processed. Each of these modules occupies 1 word; i.e., together they occupy 2 words. Counting for the

output process image begins anew at “0”. The hardware output addresses are then **%QW0** and **%QW1**.

Now the digital outputs (DO) are dealt with. These occupy 2 bits. Two complete words have been previously counted (Word 0 and 1). Now, counting is continued from Word 2 and 2 bits are added (Bit 0 and Bit 1). The hardware addresses are then **%QX2.0** and **%QX2.1**.

Note



Hardware changes can result in changes of the process image!

Changing or adding of digital, analog or complex modules (DALI, EnOcean, etc.) may result in a new process image being generated. The process data addresses would then also be changed. Therefore, the process data of all previous modules has to be taken into account when modules are added.

7.3.2.3 IEC-61131-3 Address Areas

Table 44: IEC-61131-3 Address Areas

Address area	MODBUS Access	PLC Access	Description
phys. inputs	read	read	Physical inputs (%IW0...%IW255 und %IW512...%IW1275)
phys. outputs	read/write	read/write	Physical outputs (%QW0...%QW255 und %QW512...%QW1275)
MODBUS/TCP PFC-IN variables	read/write	read	Volatile PLC input variables (%IW256...%IW511)
MODBUS/TCP PFC-OUT variables	read	read/write	Volatile PLC output variables (%QW256...%QW511)
Configuration register	read/write	-	see Section “Fieldbus Communication” > “Configuration Registers”
Firmware register	read	-	see Section “Fieldbus Communication” > “Firmware Information Registers”
Marker variables (retain)	read/write	read/write	Remanent memory (%MW0...%MW8191)

7.3.2.4 Absolute Addressing

Direct presentation of individual memory cells (absolute addresses) based on IEC-61131-3 is performed using character strings:

Table 45: Absolute Addressing

Position	Prefix	Designation	Comment
1	%	Introduces an absolute address	
2	I Q M	Input Output Flag	
3	X* B W D	Single bit Byte (8 bits) Word (16 bits) Doubleword (32 bits)	Data width
4		Address	

such as word-by-word: %QW27 (28th word), bit-by-bit: %IX1.9 (10th bit in the 2nd word)

* The designator “X” for bits can be omitted



Note

Enter character strings without spaces or special characters!

The character strings for absolute addresses must be entered connected, i.e. without spaces or special characters!

Addressing example:

Table 46: Addressing Example

Inputs			
Bit	%IX14.0 ... 15		%IX15.0 ... 15
Byte	%IB28	%IB29	%IB30 %IB31
Word	%IW14		%IW15
Double word	%ID7		

Outputs			
Bit	%QX5.0 ... 15		%QX6.0 ... 15
Byte	%QB10	%QB11	%QB12 %QB13
Word	%QW5		%QW6
Double word	%QD2 (top section)		%QD3 (bottom section)

Flags			
Bit	%MX11.0 ... 15		%MX12.0 ... 15
Byte	%MB22	%MB23	%MB24 %MB25
Word	%MW11		%MW12
Double word	%MD5 (top section)		%MD6 (bottom section)

Calculating addresses (as a function of the word address):

Bit address: Word address .0 to .15

Byte address: 1st byte: 2 x word address

2nd byte: 2 x word address + 1

DWord address: Word address (even number) / 2

or Word address (uneven number) / 2, rounded down

7.3.3 Data Exchange between MODBUS/TCP Master and I/O Modules

Data exchange between the MODBUS/TCP Master and the I/O modules is conducted using the MODBUS functions implemented in the controller by means of bit-by-bit or word-by-word reading and writing routines.

There are 4 different types of process data in the controller:

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

Access by word to the digital I/O modules is carried out in accordance with the following table:

Table 47: Allocation of Digital Inputs and Outputs to Process Data Words in Accordance with the Intel Format

Digital inputs/ outputs	16.	15.	14.	13.	12.	11.	10.	9.	8.	7.	6.	5.	4.	3.	2.	1.
Process data word	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte	High byte D1								Low byte D0							

Output can be read back in by adding an offset of 200_{hex} (0x0200) to the MODBUS address.

Note



Data > 256 words can be read back by using the cumulative offset!

All output data greater than 256 words and, therefore located in the memory range 0x6000 to 0x62FC, can be read back by adding an offset of 1000_{hex} (0x1000) to the MODBUS address.

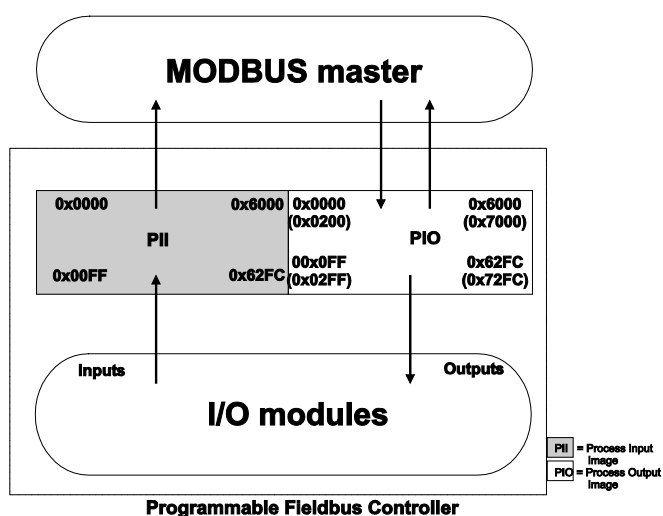


Figure 43: Data Exchange Between MODBUS Master and I/O Modules

Register functions start at address 0x1000. These functions can be addressed in a similar manner with the MODBUS function codes that are implemented (read/write).

The specific register address is then specified instead of the address for a module channel.



Information

Additional Information

A detailed description of the MODBUS addressing may be found in Chapter "MODBUS Register Mapping".

7.3.4 Data Exchange between PLC Function (CPU) and I/O Modules

The PLC function (CPU) of the PFC uses direct addresses to access the I/O module data.

The PFC uses absolute addresses to reference the input data. The data can then be processed internally in the controller using the IEC-61131-3 program. Flags are stored in a non-volatile memory area in this process. The results of linking can then be written directly to the output data employing absolute addressing.

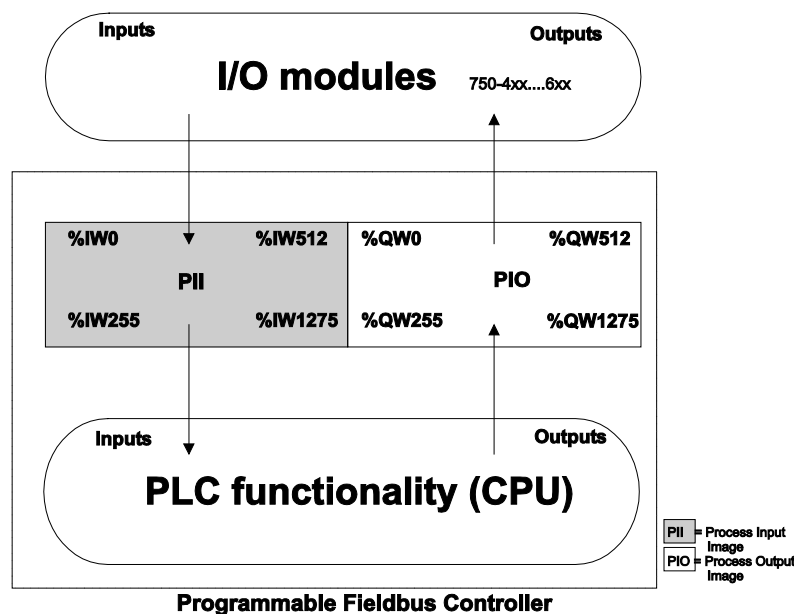


Figure 44: Data Exchange Between PLC Function (CPU) of the PFC and the I/O Modules

7.3.5 Data Exchange between Master and PLC Function (CPU)

The fieldbus master and the PLC function (CPU) of the PFC have different perspectives on data.

Variable data generated by the master are routed as input variables to the PFC, where they are further processed.

Data created in the PFC are transmitted via fieldbus to the master as output variables.

In the PFC, access to the MODBUS/TCP PFC variable data is possible starting from word address 256 to 511 (double-word address 128-255, byte address 512-1023), while access to the PLC variable data is possible starting from a word address of 1276 to 1531 (double-word address 638-765, byte address 2552-3063).

7.3.5.1 Example of MODBUS/TCP Master and PLC Function (CPU)

Data access by the MODBUS/TCP Master

Access to data by the MODBUS Master is always either by word or by bit. Addressing of the first 256 data words by the I/O modules begins with word-by-word and bit-by-bit access at 0.

Addressing of the data by the variables begins at 256 for word-based access; bit-by-bit access then takes place starting at:

4096 for bit 0 in word 256
4097 for bit 1 in word 256
...
8191 for bit 15 in word 511.

The bit number can be determined using the following equation:

$$\text{Bit No.} = (\text{word} * 16) + \text{Bit No. in word}$$

Example: 4097 = (256 * 16) + 1

Data Access by PLC Function (CPU)

The PLC function of the PFC employs a different type of addressing for accessing the same data. PLC addressing is identical with word-by-word addressing by the MODBUS Master for the declaration of 16-bit variables. However, a different notation is used for declaration of Boolean variables (1 bit) than that used by MODBUS. Here, the bit address is composed of the elements word address and bit number in the word, separated by a decimal point.

Example:

Bit access by MODBUS to bit number 4097 => Bit addressing in the PLC
<Word No.>.<Bit No.> = 256.1

The PLC function of the PFC can also access data by bytes and by doubleword access.

Addresses are calculated based on the following equations for byte-based access:

$$\text{High Byte address} = \text{Word address} * 2$$

$$\text{Low Byte address} = (\text{Word address} * 2) + 1$$

Addresses are calculated according to the following equation for double-word based access:

$$\text{Double-word address} = \text{High word address} / 2 \text{ (rounded down)}$$

$$\text{or} = \text{Low word address} / 2$$



Information

Additional Information

There is a detailed description of the MODBUS and the corresponding IEC 61131 addressing in section “MODBUS Register Mapping”.

7.3.6 Application Example

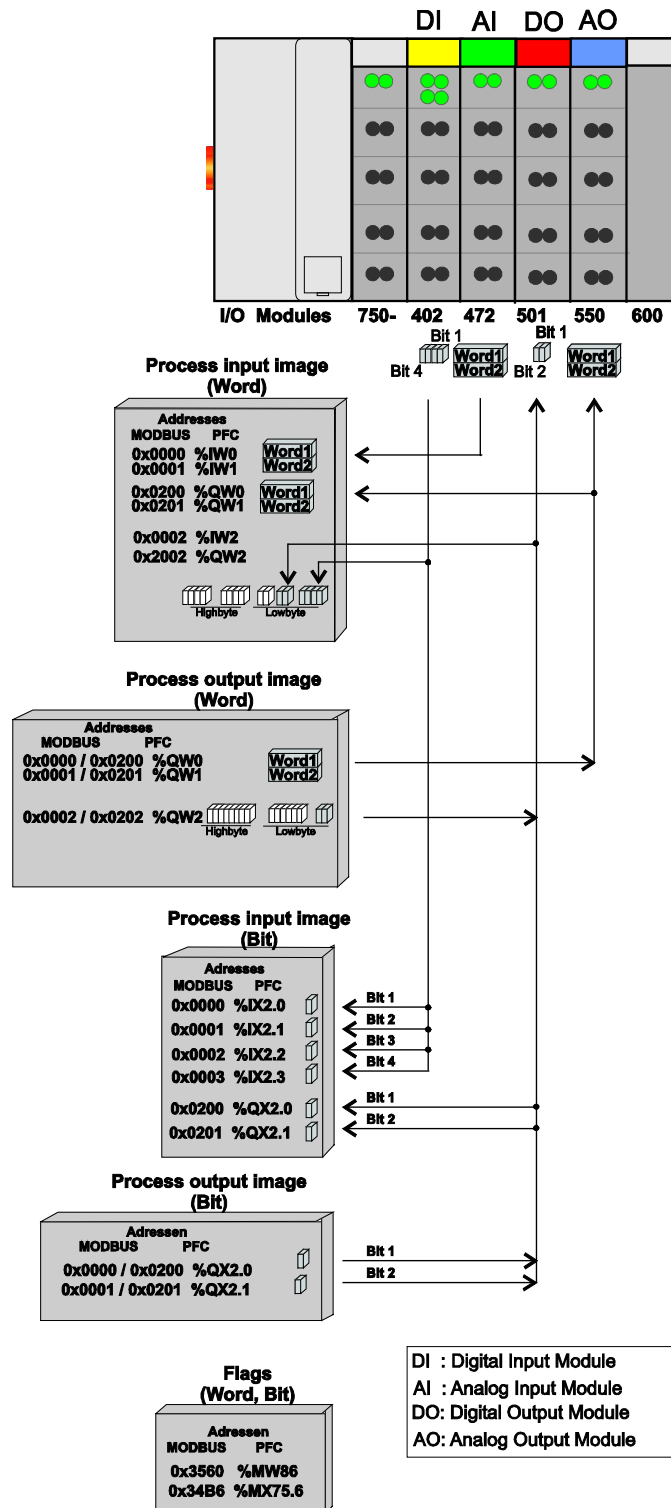


Figure 45: Example of Addressing for a Fieldbus Node

7.4 Memory Card Function

The memory card is optional and serves as memory in addition to the internal memory in the fieldbus controller. The user program, user data, the source code of the project or device settings can be saved on the memory card and thus already-existing project data and programs can be copied to one or more fieldbus controller(s).

Note

**Only use recommended memory cards!**

Use only the SD memory card available from WAGO (item no. 758-879/000-001) since it is suitable for industrial applications under difficult environmental conditions and for use in the fieldbus controller. Compatibility with other commercially available storage media cannot be guaranteed.

If the memory card has been inserted, it is incorporated into the directory structure of the internal file system of the fieldbus controller as drive S:. This way, the memory card can be addressed like a changeable medium on a PC.

Note

**Deactivate write protection!**

In order to be able to write data on the memory card, you must deactivate the small push switch for the write protection setting. This switch is on one of the long sides of the memory card.

The function of the memory card in normal operation and possible errors that can arise when using the memory card will be described in the following sections for various steps during operation:

- Project copying and automatic system start
- Back-up function (Storing device-internal data and settings on the memory card)
- Restore function (Loading device-internal data and settings from the memory card)
- Saving of BACnet trend data
- Inserting a memory card during ongoing operation
- Removing the memory card during ongoing operation
- Saving WAGO-I/O-PRO project to memory card
- FTP network access to the file system of the memory card
- Directory structure



Note

Note the pre-formatting of SD card!

Note that SD cards ≤ 2 GB are often formatted with the file system type “FAT16”, and the creation of a maximum of 512 entries in the root directory is possible. For more than 512 entries, create them in a subdirectory, or format the SD card with “FAT32”.

7.4.1 Project Copying and Automatic System Start

This function is useful for project updates or project copying between several fieldbus controllers using an SD card.



Note

Replication only possible between fieldbus controllers of the same type!
Make sure to execute the “Restore” function for replication between fieldbus controllers with identical item numbers only.

First, start the fieldbus controller without the SD card being inserted.

The “Autorestore Global Enable” function is enabled in the Web-based Management on the “Backup & Restore” website. In this context, choose to restore after each reboot or in this case, to replicate (“Always restore settings from disk on reboot”) or to restore once on the next reboot only (“Restore settings from disk on next reboot only”).

Restore Option Setting	
Setting	Description
☒	Autorestore Global Enable This option will be disabled after the next reboot if the following option is set to next reboot only
☒	Restore settings from disk on next reboot only
☐	Always restore settings from disk on reboot

Figure 46: Restore Function

Select the data to be copied from the SD card of another fieldbus controller in the “Settings” column. Only selected options are transferred (e.g., device settings, ports, CODESYS settings, file system, etc.).

Power supply is interrupted, allowing you to insert the SD card from a different fieldbus controller.

Then switch on the power for the fieldbus controller.

The fieldbus controller is started and the data selected in “Settings” is copied from the S:/copy/xxx directories on the memory card to the respective directories of the internal file system.

Identical data that is already present in the internal file system is not copied.

The LED above the SD card flashes while copying is in progress.

The copied data is then loaded automatically. Device settings and other data are saved in the fieldbus controller and are executed.

Note



Disable automatic restore function when not used!

If the “Autorestore Global Enable” function, as well as the associated “Always restore settings from SD card on reboot” or “Restore settings from disk on next reboot only” functions are enabled, then the fieldbus controller is checked each time or on the next reboot if there is already a back-up on the SD card (“copy/xxx” directory). If there is a back-up, the selected parts of the back-up are loaded from the SD card to the fieldbus controller (Restore). The internal file system is written using the data from the SD card.

Attention: Any boot project in the fieldbus controller is deleted. Disable the “Autorestore Global Enable” function if you do not want the automatic restore function to execute on each or the next reboot.

Table 48: Possible Errors on System Start

Possible error	Procedure	Error message	Solution
No /copy/ and /settings/ directories on the memory card	System start of the fieldbus controller is running as if no memory card were present.	No message.	If memory card use is desired, map the directory structure of the internal drive under /copy/ on the memory card on the PC. Then restart the fieldbus controller.
Manual interrupt	If the /copy/ directory has not yet been copied in full, then the copy process is stopped on manual interrupt. The data is incomplete.	Blink code of the I/O LED for 30 seconds (error code 14, error argument 2) or until the error is eliminated.	Restart system.
Internal file system is full	As much data is copied until the file system is full. The copy function is cancelled. The data is incomplete.	Blink code of the I/O LED (error code 14, error argument 3) until the error is eliminated.	Check internal file system and /copy/ directory on the memory card and delete data if necessary (e.g. via FTP, formatting internal drive or cleaning the memory card on the PC).
General internal error	The copy function is cancelled. The data is incomplete.	Blink code of the I/O LED (error code 14, error argument 7).	If necessary, restart function and device.

Note



The file size may not be larger than the size of the internal drive!

Note that the amount of data in the /copy/ directory must not exceed the total size of the internal drive when restoring.

7.4.2 Back-up Function (Storing Device-internal Data and Settings on the Memory Card)

The fieldbus nodes and the SPS program are in operation and the memory card is plugged in.

On the “Backup & Restore” page in the Web-based management system start the function “Backup ALL device settings to removable disk” to save the data on the internal drive and the device settings to the memory card during operation.

Services		
Command	Description	Status
START	Backup ALL device settings to removable disk (77 kB needed).	OK
START	Restore device settings from removable disk using the option settings below.	OK

Figure 47: WBM Page “Backup & Restore”

In this process, the files from the internal drive are stored in the directory S:/copy/xxx and in the corresponding subdirectories on the memory card. The information that is not present as files on the fieldbus controller is stored in XML format in the S:/settings/ directory.

Meanwhile, the SD-LED blinks yellow/orange.

The device settings and files of the internal drive are then saved on the memory card.

Table 49: Possible Errors During the Backup Function

Possible error	Procedure	Error message	Solution
Memory card is full	The copy function is cancelled. Device settings and files in the /copy/ directory and the subdirectories are not saved completely on the memory card.	Blink code of the I/O LED (error code 14, error argument 4) until the error is eliminated.	Replace the memory card with a new one, then restart the backup function. Create storage space on the memory card with the PC.
Memory card is write-protected.	Copying is not possible, the backup function is cancelled and the memory card remains as it was.	Blink code of the I/O LED (error code 14, error argument 6) until the error is eliminated.	Remove the memory card, remove the write protection, insert the card again, then restart the backup function.
Manual interrupt	The copy function is cancelled. Device settings and files in the /copy/ directory and the subdirectories are not saved completely on the memory card.	Blink code of the I/O LED for 30 seconds (error code 14, error argument 2) or until the error is eliminated.	Insert memory card, then restart the backup function.
General internal error	The copy function is cancelled. Device settings and files in the /copy/ directory and the subdirectories are not saved completely on the memory card.	Blink code of the I/O LED (error code 14, error argument 6).	If necessary, restart function and device.

7.4.3 Restore Function (Loading Device-internal Data and Settings from the Memory Card)

The fieldbus nodes and the SPS program are in operation and the memory card is plugged in.

On the “Backup & Restore” page in the Web-based management system, start the “Restore device settings to removable disk using the options below” function in order to load the device settings from the memory card to the internal drive during normal operation.

Services		
Command	Description	Status
START	Backup ALL device settings to removable disk (77 kB needed).	OK
START	Restore device settings from removable disk using the option settings below.	OK

Figure 48: WBM Page “Backup & Restore”

For the Restore function, other options can be set that are described in the chapter “Configuring in the Web-based Management System (WBM)”.

Restore Option Setting	
Setting	Description
☐	Autorestore Global Enable This option will be disabled after the next reboot if the following option is set to next reboot only
☒	Restore settings from SD card on next reboot only
☐	Always restore settings from SD card on reboot
☒	IP address, dip switch IP address, network mask, domain name, host name, gateway, DNS servers, auto reset on system error, BOOTP request before static IP, non-adaptive Kbus speed
☒	Ports, security, time server, multicast settings, blocked Modbus flag registers, MAC filter, Ethernet hardware settings
☒	PLC settings, PLC folder and subfolders, boot project, ETC folder and web visualisation
☒	Non-volatile memory
☒	File system
☒	Watchdog, trace settings, SNMP, Ethernet/IP padding, clock settings

Figure 49: WBM Page “Backup & Restore”

In this process of loading device settings from the Memory Card, the data are copied from the directory S:/copy/ on the memory card to the appropriate directories on the internal drive.

The SD LED flashes yellow/orange while this is in progress.

Note



In case of parameter changes, the device executes a restart!

Note that the device taking the data executes a restart if parameters in the internal drive are overwritten with other parameter settings from the memory card.

The boot project is loaded automatically and the settings automatically activated after a restart. The “PLC ROOT Location” defined on the “PLC” Web page determines whether the boot project of the internal drive or the SD card is loaded.

Table 50: Possible Errors During the Restore Function

Possible error	Procedure	Error message	Solution
No /copy/ and /settings/ directories on the memory card	Fieldbus controller is working as if no memory card were present.	No message.	If memory card use is desired, map the directory structure of the internal drive under /copy/ on the memory card on the PC. Then restart the fieldbus controller.
Manual interrupt	If the /copy/ directory has not yet been copied in full, then the copy process is stopped on manual interrupt. The data is incomplete.	Blink code of the I/O LED for 30 seconds (error code 14, error argument 2) or until the error is eliminated.	Restart system.
Internal file system is full	As much data is copied until the file system is full. The copy function is cancelled. The data is incomplete.	Blink code of the I/O LED (error code 14, error argument 3) until the error is eliminated.	Check internal file system and /copy/ directory on the memory card and if necessary, delete files (e.g. through FTP, formatting internal drive or cleaning the memory card on the PC).
General internal error	The copy function is cancelled. The data is incomplete.	Blink code of the I/O LED (error code 14, error argument 7).	If necessary, restart function and device.

Note



The file size may not be larger than the size of the internal drive!

Note that the amount of data in the S:/copy/ directory must not exceed the total size of the internal drive when restoring.

7.4.4 Saving BACnet Trend Data

A BACnet trend log object can monitor and detect properties of referenced objects and save them with a time flag. These trend logs are normally stored on the internal drive of the fieldbus controller.

We recommend saving the BACnet trend data on the SD card in order to enhance storage capacity for extensive trend logs.

To do this, select the option “Save trendlog files on removable disk” on the “BACnet” Web page in the Web-based management system to define the SD card as the storage location. Existing trend logs are retained on the internal drive of the fieldbus controller. New trend logs will be stored on the SD card after this setting is defined.

Configuration	
Disable BACnet fieldbus	☐
UDP Port	47808
Who-Is online interval time (seconds)	60
Non-adaptive terminal bus speed	☒
Save trendlog files on removable disk	☐
Disable PLC online change	☒

Figure 50: WBM Page “BACnet”

If the SD card is removed even though it is activated as the storage location and the fieldbus coupler/controller is in normal operation, this setting is retained in the WBM. Saving of trend logs is then discontinued. The BACnet/IP protocol and WAGO-I/O-PRO both continue to run, however. The BACnet property “Reliability” for the trend log object, when available, is then set to “UNRELIABLE_OTHER”. No flash code is output in this case.

Saving of trend logs is resumed when the SD card is re-inserted.



Note

Note limits when creating trend log objects!

To avoid filling the internal drive or SD card with BACnet trend data completely, the data volume is limited for this purpose.

The upper limit is:

- 60,000 trend logs when using the internal memory
- 4,000,000 trend logs when using an SD card

If the limit is exceeded, no further trend log objects can be created. No error is displayed.

7.4.5 Inserting a Memory Card During Operation

The fieldbus nodes and the SPS program are running.

Insert a memory card during ongoing operation.

During normal operation, the memory card is incorporated into the file system of the fieldbus controller as a drive.

No automatic copy procedures are triggered.

The SD-LED blinks yellow/orange during the access.

The memory card is then ready for operation and available as drive S:.



Note

BACnet functionality after re-inserting the SD card not active until after a restart!

If the “PLC Root Location” on the WBM page “PLC” is set to an external location (saving to the SD card), inserting the SD card during operation has no effect on the BACnet functionality. Not until the next restart of the fieldbus coupler/controller is the configuration loaded from the SD card and used.

Table 51: Possible Errors when Inserting the Memory Card During Ongoing Operation

Possible error	Procedure	Error message	Solution
Access to the memory card is not possible	The incorporation (mounting) of the memory card as drive is cancelled. The fieldbus controller behaves as if no memory card were present.	Blink code of the I/O LED (error code 14, error argument 1) until the error is eliminated.	Replace the memory card with a new one.
Memory card cannot be read	The incorporation (mounting) of the memory card as drive is cancelled.	Blink code of the I/O LED (error code 14, error argument 1) until the error is eliminated. For access through SPS, the error is reported via the CODESYS library.	Replace the memory card with a new one.
Memory card is not formatted	The incorporation (mounting) of the memory card as drive is cancelled.	Blink code of the I/O LED (error code 14, error argument 1) until the error is eliminated. For access through SPS, the error is reported via the CODESYS library.	Replace memory card with a new one or format with PC.
Memory card is not in FAT or FAT32 format.	The incorporation (mounting) of the memory card as drive is cancelled.	Blink code of the I/O LED (error code 14, error argument 1) until the error is eliminated. For access through SPS, the error is reported via the CODESYS library.	Replace memory card with a new one or format with PC.

7.4.6 Removing the Memory Card During Operation

The fieldbus nodes and the SPS program are in operation and the memory card is plugged in.

Remove the memory card during ongoing operation.

Note



Data can be lost during writing!

The SD-LED blinks yellow/orange during the access.

Note that if you pull the memory card out during a write procedure, data will be lost. Even persistent values for BACnet properties, objects added online or trend log entries can be lost.

The fieldbus controller then works without memory card.

Table 52: Possible Errors when Removing the Memory Card During Ongoing Operation

Possible error	Procedure	Error message	Solution
Read procedure is running	Read access is cancelled. The fieldbus controller then works without memory card.	Blink code of the I/O LED for 30 seconds (error code 14, error argument 2) if the backup or restore function is active. For access through SPS, the error is reported via the CODESYS library.	If memory card usage is desired, insert the memory card, then restart the access function.
Write procedure is running	Write access is cancelled. The fieldbus controller then works without memory card.	Blink code of the I/O LED for 30 seconds (error code 14, error argument 2) if the backup or restore function is active. For access through SPS, the error is reported via the CODESYS library.	If memory card usage is desired, insert the memory card, then restart the access function.

7.4.7 Saving WAGO-I/O-PRO project to Memory Card

To save a WAGO-I/O-PRO project on the SD card and to have it executed from that location, activate the option “External SD memory card” on the WBM page “PLC” (see also the Section “Configuration in the Web-based Management-System” > “PLC page”).

This setting is remanent. Normally, “Internal File System” is activated.

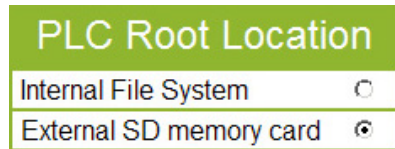


Figure 51: Setting the Storage Location in the Web-based Management System

Switching of the PLC root location occurs immediately when you click the button **[SUBMIT]**.

If no “PLC” directory exists on the SD card, one is created.
If one exists, it is neither deleted nor overwritten.

WAGO-I/O PRO will then access the SD card file system (S:/PLC) instead of the internal drive.

In this case, data (e.g., retain data) are read/written to or from the memory card, or copied from/to the memory card file system.

In normal operation, data is read into the file system of the memory card or read from it for further processing by the IEC-61131 application via a firmware library.

The LED above the SD card flashes yellow/orange while this is in progress.

The data is then written or read.

The BACnet configuration of the new root is used when the "PLC Root Location" setting is changed.



Note

Redefine the boot project after changing the "PLC Root Location"!

If you have changed the "PLC Root Location", the fieldbus controller boot project will not be started, as it is then not located at the current "PLC Root Location".

No old data will be copied to the new "PLC Root Location".

You must therefore generate a new boot project using the menu item "Create boot project" in WAGO-I/O-PRO.

If you require data for this which were created at the old "PLC Root Location", copy these manually into the new root directory.

Note



Provide sufficient storage space for PLC folders!

The following can occur when there is less free storage space on the SD card than is normally required for the PLC folder and the files it contains:

- The files "webvisu.htm" and "error_ini.xml" are created without any content, meaning that the WBM page "Web Visu" is blank.
- The files "webvisu.htm" and "error_ini.xml" are created but are incomplete.

In either case, this condition remains until the entire "PLC" folder is deleted and sufficient storage space made available.

The function "PLC Root Location" remains set to "External SD memory card".

The MS/BT LED flashes red.

Note



BACnet file objects can be created internally and on the SD card!

BACnet file objects can be created both on the internal drive and on the SD card. The creation location is indicated by the root designator (A: (internal) or S: (SD card)) in the name. If no root is indicated in the name, the drive is determined from the defined "PLC Root Location" (WBM page "PLC").

Note



Log in to use the PLC browser with administrator user data!

To use the PLC browser functionality in WAGO-I/O-PRO, log into the PLC browser with the administrator user data (default user "admin", password "wago"). Enter "login admin wago" in the command line of the PLC browser.

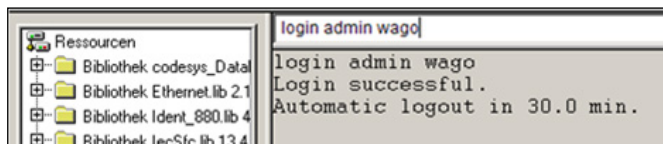


Figure 52: Logging into the PLC Browser

Note



Use proper path for Web pages when saving to SD card!

Web pages on the SD card can be accessed by entering the following path in the browser URL line:

[http://\[IP address\]/SD/\[subdirectory structure on SD card\]](http://[IP address]/SD/[subdirectory structure on SD card])

Note



Save the PLC program and Web visualization on same drive!

If you are using a PLC program and Web visualization, it is imperative that you have both of these items on the same drive, as otherwise Web visualization may not start.

Table 53: Possible Errors During SPS Access to the File System of the Memory Card

Possible error	Procedure	Error message	Solution
Memory card is full	Read functions are executed, in case of write access, the function is cancelled. The data is incomplete or not updated.	Error is reported via the CODESYS library.	Replace the memory card with a new one, then restart the access function. Make space on the memory card with the PC.
Memory card is write-protected.	Read functions are executed, write access is not possible. The memory card remains as it was.	Error is reported via the CODESYS library.	Remove the memory card, remove the write protection, insert the card again, then restart the access function.
No memory card inserted.	The BACnet stack does not start up. No access to BACnet functions.	Error is reported via the CODESYS library. Flash code 10-1 (error during initialization of the runtime system) output by the fieldbus controller.	If memory card usage is desired, insert the memory card, then restart the access function.
Manual interrupt	Access function is cancelled.	Error is reported via the CODESYS library.	Restart the access function.

7.4.8 FTP Network Access to the File System of the Memory Card

The fieldbus node is in operation and the memory card is plugged in.

The FTP client accesses the file system of the memory card (drive S:) via the network.

In normal operation, the FTP client writes data to the file system of the memory card or reads data from the file system of the memory card.

Meanwhile, the SD-LED blinks yellow/orange.

The data is then written or read.

Table 54: Possible Errors During the FTP Network Access on the File System of the Memory Card

Possible error	Procedure	Error message	Solution
No memory card is plugged in.	Access function is cancelled.	Depending on the FTP client used, the error is reported.	If memory card usage is desired, insert a memory card, then if necessary, close the existing FTP connection and establish a new one, then restart FTP access.
Manual interrupt	Access function is cancelled.	Depending on the FTP client used, the error is reported.	If necessary, close existing FTP connection and establish a new one, then restart FTP access.

Note



Observe FTP access rights!

The FTP access rights correspond to those for the WBM. Note here that with “guest” access rights you are provided only with read access to the file system. Log in as “admin” to obtain write access.

Note



FTP connection must be closed and then re-established!

Note that if you pull the memory card out with an existing FTP connection and then insert it again, that you must then close the existing FTP connection and establish it again so that access to the memory card is possible.

7.4.9 Directory Structure

On the WBM page "Features", you can create a backup copy of the internal drive using the function "**Backup** device settings to removable disk". The folder "copy" is created for this on the SD card. All of the files and subdirectories on the internal drive "A:" will then be copied to this folder.

In addition, the device settings are also written to the directory "S:\settings" on the memory card.

On the WBM page "Features" you can restore the SD card data to the internal drive "A:" using the function "**Restore** device settings from removable disk". All of the files and subdirectories are copied for this from the directory "S:/copy" on the memory card to the internal drive "A:".

In addition, the device settings are loaded from the directory "S:\settings" and activated.

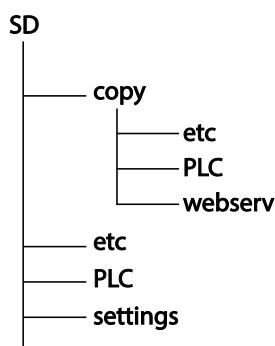


Figure 53: Example of Directory Structure on the SD Card

Table 55: Data Saved in the Directory Structure

Directory	Data saved
S:\copy	The folder "copy" is created on the SD card via the backup function of the Web-based Management System (WBM page "Feature"). This folder contains a copy of the internal drive.
S:\copy\etc	Node configuration, BACnet files (backup)
S:\copy\PLC	Project files (backup) A boot project is started, provided the "PLC Root Location" is set to "internal" in the WBM.
S:\copy\webserv	WBM Web pages (backup)
S:\settings	All saved EEPROM parameters (backup)
If saving to an "External SD memory card" is activated in the WBM on the page "PLC", the following directories will also be created in the SD card directory:	
S:\etc	BACnet configuration files (SYM_XML, Override, etc.)
S:\PLC	Project files A boot project is started, provided the "PLC Root Location" is set to "external/SD" in the WBM.

8 Commissioning

This section shows a step-by-step procedure for starting up exemplarily a WAGO fieldbus node.

Note



Exemplary Example!

This description is exemplary and is limited here to the execution of a local start-up of one individual fieldbus node with a non-interlaced computer running Windows.

For start-up, three steps are necessary. The description of these work steps can be found in the corresponding following sections.

- **Connecting PC and Fieldbus Node**
- **Allocating the IP Address to the Fieldbus Node**
- **Testing the Function of the Fieldbus Node**

Following the commissioning descriptions after which the fieldbus node is ready for communication, the following topics are described:

- **Preparing the Flash File System**
- **Synchronizing the real-time clock**
- **Restoring factory settings**

After the topics specified above, you can find instructions for programming the fieldbus controller with WAGO-I/O-PRO and the description of the internal web pages of the web-based Management System (WBM) for additional settings of the fieldbus controller.

8.1 Connecting Client PC and Fieldbus Nodes

1. Mount the fieldbus node to the carrier rail.
Observe the installation instructions described in "Mounting" section.
2. Connect the 24V power supply to the supply module.
3. Connect an ETHERNET interface from the client PC to an ETHERNET interface of the fieldbus controller.
4. Turn the operating voltage on.
Make sure that the mode selector is not in the bottom position.

After the power is switched on, the fieldbus controller is initialized. The fieldbus controller determines the I/O module configuration and creates a process image. During startup, the I/O LED (red) will flash. After a brief period, the I/O LED lights up green, indicating that the fieldbus controller is operational.

If an error has occurred during startup, a fault code is flashed by the I/O LED. If the I/O LED flashes 6 times (indicating error code 6) and then 4 times (indicating error argument 4), an IP address has not been assigned yet.

8.2 Allocating the IP Address to the Fieldbus Node

There are various ways to assign the IP address.
The various options are described in the following sections individually.

- Use **address selection switch** (DIP switch) to assign IP address (manually).
- **Assigning IP Address via WAGO ETHERNET Settings**
(static IP address via the serial communication port)
- **Assigning IP Address via PLC program**
(Static IP address)
- **Assigning IP Address via BootP server**
(Static IP address via the fieldbus)



Note

The IP address must occur in the network only once!

For error-free network communication, note that the assigned IP address must occur only once in the network!

In the event of an error, the error message "IP address configuration error" (error code 6 – error argument 7) is indicated by 'I/O' LED at the next power-on.



Note

Check the network for a DHCP server when using DHCP!

If you are using DHCP, then make sure that there is a DHCP server in the network and that an IP address can be assigned to the fieldbus controller before start-up. BACnet is not initialized without an IP address and no application is launched.

8.2.1 Information about the Starting Behavior regarding BACnet and WAGO-I/O-PRO Program

The address selector switch is set to 0 by default, i.e. the IP configuration from the Web-based Management system (WBM) is used. The IP address assignment is again set to BootP by default in the WBM.

Please note the following regarding BACnet functions and execution of a loaded WAGO-I/O-PRO program:

1. If **BootP** is set (default) and the "BootP Request before Static-IP" parameter is not set in the WBM on the "Features" page (default), BACnet and the WAGO-I/O-PRO program only start if an address has been assigned by the BootP server.
2. If **BootP** is set and the "BootP Request before Static-UP" is set in the WBM on the "Features" page, the BACnet function is enabled after a timeout period and the WAGO-I/O-PRO program also started without a BootP server. There must be an IP address in the EEPROM that has been manually assigned or previously assigned by a BootP server.
3. If you use a **static IP address** (setting via ETHERNET settings), the BACnet functions are immediately enabled and the WAGO-I/O-PRO program started.
4. If **DHCP** is set (e.g. via address selector switch 255 or address selector switch 0 with enabled DHCP in the WBM), an IP address must be automatically assigned by a DHCP server in the network. Only after the IP address is assigned are the BACnet functions enabled and then an existing WAGO-I/O-PRO started.

8.2.2 Assigning IP Address via Address Selection Switch

Use the address selection switch to set the host ID, i.e., the last byte of the IP address, which is entered in the Web-Based Management System on WBM page “TCP/IP”, entry “DIP switch IP-Address”, with values between 1 and 254 binary coded.

Example:

IP address saved in the fieldbus controller:	192.168.1
Set DIP switch value:	50 (00110010 _{bin})
Resulting IP address saved in the fieldbus controller:	192.168.1. 50

Note



Host ID 1 - 254 via address selection switch freely adjustable!

Use the address selection switch to set the last byte of the IP address to a value between 1 and 254. The DIP switch is then enabled and the IP address is composed of the DIP switch base address stored in the fieldbus controller and the host ID set on the DIP switch. The IP address make via the Web-based Management System or WAGO-ETHERNET-Settings is disabled.

Note



Address selection switch values 0 and 255 are predefined, address selection switch disabled!

If you use the address selection switch to set the value 0 or 255, the address selection switch is disabled and the setting configured in the fieldbus controller is used. With the value 0, the settings of the Web based Management System or ETHERNET-Settings apply. If you set the value 255, the configuration via DHCP is activated.

The base address used consists of the first three bytes of the IP address. This always depends on the IP address currently saved in the fieldbus controller. The address selection switch setting to 1 ... 254 then overwrites accordingly the value of the host ID.

If there is still no static IP address in the fieldbus controller, the default value **192.168.1**. defined by the firmware as the base address is used when setting the DIP switch to 1 ... 254.

Information



More information about changing the static base address

You can also change the base address currently saved in the fieldbus controller as required. Proceed as described in the following section "Assigning IP Address via WAGO ETHERNET Settings".

1. To configure the IP address via the address selection switch by setting the host ID (last position of the IP address) to a value that does not equal 0/255,

first convert the host ID to the binary representation.

For example, host ID 50 results in a binary code of 00110010.

2. Set the bits in sequence using the 8 address switches. Start with address switch 1 to set bit 0 (LSB) and end with address switch 8 for bit 7 (MSB).

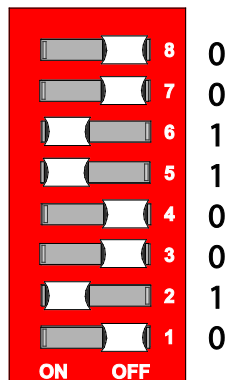


Figure 54: Address Selection Switch, for Example the Value Setting “50” ($2^1 + 2^4 + 2^5$)

3. Restart the fieldbus coupler after adjusting the address selection switch to apply the configuration changes.

8.2.3 Assigning IP Address via WAGO-ETHERNET-Settings

This program is used to configure an IP address via serial communications interface, to reset the fieldbus coupler parameters to the factory settings and to delete and extract the Flash File System in which the HTML pages of the fieldbus coupler are stored.

WAGO communication cables or WAGO radio-link adapters can be used for data communication.

NOTICE

Do not connect Communication Cable when energized!

To prevent damage to the communications interface, do not connect or disconnect Communication Cable 750-920 respectively 750-923 when energized!

The fieldbus controller must be de-energized!

Note

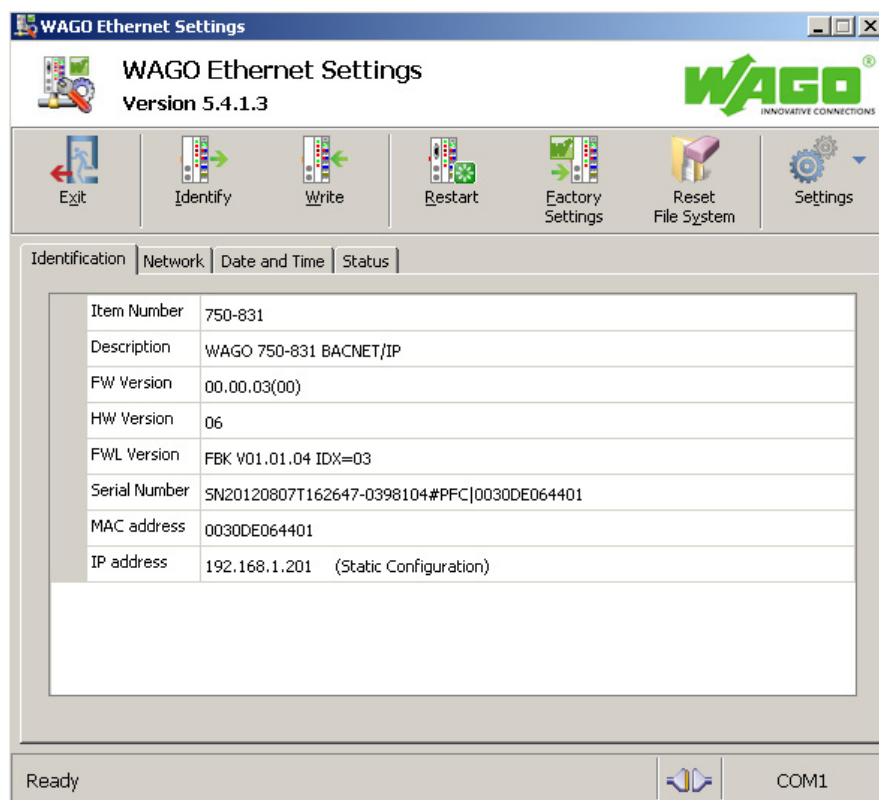


Set the address selection switch to 0 for active software configuration!

Set the address selection switch to 0 to disable the DIP switch and to enable the software configuration via ETHERNET-Settings.

Restart the fieldbus node after adjusting the switch to apply the configuration changes.

1. Using a WAGO Communication cable 750-920 respectively 750-923, connect your PC with the service interface of the fieldbus controller.
2. Start **WAGO-ETHERNET-Settings** program.
3. Click on **[Read]** to read in and identify the connected fieldbus node.
4. Select the **Network** tab.



5. To assign a permanent address, select the option "Static configuration" in the field **Source** (BootP is the default).
6. Enter the required **IP Address** and, if applicable, the address of the subnet mask and gateway.
7. Click on the **[Write]** button to apply the settings in the fieldbus node.
8. You can now close ETHERNET-Settings or make other changes in the Web-based Management System as required. To open the Web-based Management System click on the button **[WBM]** on the right side.

8.2.4 Assigning the IP Address with a PLC program

A PLC program can be used to assign a fixed IP address.

When assigning an address using a PLC program, this can be done using the "Ethernet_Set_Network_Config" function block from the "Ethernet.lib" library integrated in WAGO-I/O-PRO.

Information



Additional Information about IP address assignment via a PLC program!

A detailed description of the library for the address assignment via a PLC program, refer to the manual for the WAGO-I/O-PRO library "Ethernet.lib". This can be found on the website under <http://www.wago.com> → Downloads → Additional documentation and Information on automation products → WAGO Software → WAGO-I/O-PRO/CODESYS → Additional Information → Libraries → Ethernet.lib!

8.2.5 Assigning the IP Address with a BootP Server

A BootP server can be used to assign a fixed IP address.

Assigning the IP address using a BootP server depends on the respective BootP program. Handling is described in the respective manual for the program or in the respective integrated help texts.

Note



Set the address selection switch to 0 for active software configuration!

Set the address selection switch to 0 to disable the DIP switch.

If an access to the internal Web pages of WBM is already possible via an IP address, then the software configuration via the WBM can be done.

In the default status, configuration via BootP is activated.

If not yet been assigned IP address, you can otherwise use WAGO-Ethernet-Settings or enable BootP in the **Network** tab of WAGO-Ethernet-Settings.

Restart the fieldbus node after adjusting the address selection switch to apply the configuration changes.

Note



IP address assignment is not possible via the router!

The IP address is assigned via patch cable, switches, hubs, or via direct link using a crossover cable. Addresses cannot be assigned via routers.

Note



BootP must be enabled on the Web pages!

Note that BootP must be enabled on the internal Web pages of the WBM, HTML page "Port configuration".

BootP is enabled by default when delivered.

Note



Set static IP automatically if BootP fails!

To automatically set a static IP address if the BootP request fails, enable the "BootP Request Before Static-IP" function on the "Features" page of the WBM.

Information



Additional Information

Assigning IP addresses using a BootP server can be carried out in any Windows and Linux operating system. Any other BootP servers may also be used, besides the WAGO-BootP server.

Information



More information about the WAGO-BootP-Server

The "WAGO-BootP-Server 759-315" is available free of charge at <http://www.wago.com>.

The following steps are included:

- Note MAC ID
- Note IP address
- Assigning the IP address and enable BootP
- Assigning the IP address permanently by option "use IP from EEPROM"

8.2.5.1 Note MAC ID

1. Write down the controller's MAC address (see label or peel-off strip).
If the fieldbus is already installed, turn off the operating voltage of the fieldbus controller, then take the fieldbus controller out of the assembly of your fieldbus node and note the MAC ID of your fieldbus controller.

The MAC ID is applied to the back of the fieldbus controller or on the paper strip with two self-adhesive peel-off strips on the side of the fieldbus controller.

MAC ID of the fieldbus controller: 00:30:DE:__:__:__

2. Plug the fieldbus controller into the assembly of the fieldbus node.
3. Use the fieldbus cable to connect the fieldbus connection of your mechanically and electrically assembled fieldbus node to an open interface on your PC.

The PC must be equipped with a network card for this connection. The controller transfer rate then depends on the network card of your PC.

4. Start the PC that assumes the function of the master and BootP server.
5. Switch on the power at the fieldbus controller (DC 24 V power supply unit).

After the power is switched on, the fieldbus controller is initialized. The fieldbus controller determines the I/O module configuration and creates a process image. During startup, the I/O LED (red) will flash. After a brief period, the I/O LED lights up green, indicating that the fieldbus controller is operational.

If an error occurs during start-up indicated by the I/O LED flashing red, evaluate the error code and argument and resolve the error.



Information

More information about LED signaling

The exact description for evaluating the LED signal displayed is available in the section “Diagnostics” > ... > “LED Signaling”.

Error codes and error arguments are indicated by the frequency of a LED flash sequence. For example: Error code 6, followed by error argument 4, is indicated by the I/O LED after controller start-up with 6 red error code flashes, followed by four red flashes of the error argument. This indicates that an IP address has not yet been assigned.

8.2.5.2 Determining IP addresses

1. If the PC is already integrated into an IP network, you can determine the PC's IP address by clicking on **Control Panel** from the **Start Menu / Settings**.
2. Double-click on the **Network** icon.
The network dialog window appears.

For Windows 2000/XP:

- Select **Network and Dial-Up Connections**
- In the dialog window that then appears, right click on **LAN Connection** and open the **Properties** link.
- Mark the entry **Internet Protocol (TCP/IP)**

For Windows 7:

- Choose **Network and Sharing Center** by using Control Panel.
- In the dialog window that then appears, right click on **LAN Connection** and open the **Properties** link.
- Mark the entry **Internet Protocol V4**

Note



Reinstall TCP/IP components if required!

If the "Internet Protocol TCP/IP" entry is missing, install the corresponding TCP/IP components and reboot your computer.

You will need the installation CD for Windows 2000, XP or 7.

3. Then click on the **Properties...** button
4. The IP address, subnet mask and, where required, the PC's gateway address appear in the Properties window. Note these values:

PC IP address: _ _ _ . _ _ _ . _ _ _ . _ _ _
 Subnet mask: _ _ _ . _ _ _ . _ _ _ . _ _ _
 Gateway: _ _ _ . _ _ _ . _ _ _ . _ _ _

5. Now select the desired IP address for your fieldbus node.

Note



Assign the PC a fixed IP address and note common subnet!

Note that the PC, on which the BootP server is listed, must have a fixed IP address and that the fieldbus node and PC must be in the same subnet.

6. Note the IP address you have selected:

Fieldbus node IP address: _ _ _ . _ _ _ . _ _ _ . _ _ _

8.2.5.3 Assigning the IP address

1. Based on the handling, which depends on the BootP program set, assign the required IP address for your fieldbus node.
2. Enable the query/response mechanism of the BootP protocol based on the handling, which depends on the BootP program set. Alternatively, do the activation of BootP in WAGO-Ethernet-Settings or if an access to the internal Web pages of WBM is already possible via an IP address, in the Web-based management system.
3. To apply the new IP address, use e.g. a hardware reset to restart your fieldbus node by interrupt the voltage supply for approx. 2 seconds.

8.2.5.4 Assigning the IP address permanently by option “use IP from EEPROM”



Note

BootP server for BACnet/IP Controller is enabled standard!

By default, the IP address is assigned to the BACnet/IP Controller through a BootP server.

When the BootP protocol is activated the fieldbus controller expects the BootP server to be permanently available. If there is no BootP server available after a PowerOn reset, the network will remain inactive.

You must then deactivate the BootP protocol so that the controller uses the configured IP address from the EEPROM; this does away with the need for the BootP server to be permanently available.



Note

The IP address stored in the EEPROM must be used for permanent address assignment!

To apply the IP address obtained via BootP permanently in the fieldbus controller, you must select the option “use IP from EEPROM”.

When this option is selected, the IP address is entered and used automatically as a static address in the EEPROM.



Note

The IP address is not lost when the BootP-Protocol is disabled!

If the BootP protocol is deactivated after addresses have been assigned, the stored IP address is retained, even after an extended loss of power, or when the fieldbus controller is removed.

You can enable the option “use IP from EEPROM” in the Web-based Management System.

Disable BootP in the Web-based Management System

1. Open the **Web browser** on your client (such as the Microsoft Internet Explorer) to have the HTML pages displayed.
2. Enter the **IP address** for your fieldbus node in the address line of the browser and press **[Return]**.
3. If a dialog window then appears with a password prompt, as Administrator, enter the user name: "**admin**" and the password "**wago**".

This is provided for secure access and entails three different user groups: admin, guest and user.

A start page is then displayed in the browser window with information about your fieldbus controller (start page can be changed on HTML page "Security").

You can navigate to other information using the hyperlinks in the left navigation bar.



Note

Disable the proxy server to display the web-based Management-System!

If these pages are not displayed for local access to the fieldbus nodes, you must define in the Web browser properties that, as an exception, no proxy server are to be used for the node IP address.

4. In the left navigation bar click on **Port** to open the HTML page for selecting a protocol.

Web-based Management

Port configuration

This page is for the configuration of the network protocols. The configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

Port Settings

Protocol	Port	Enabled
FTP	21	☒
SNTP	123	☐
HTTP	80	☒
HTTPS	443	☐
SNMP	161, 162	☒
Modbus UDP	502	☒
Modbus TCP	502	☒
WAGO Services	6626	☒
CoDeSys	2455	☒
BootP	68	☐
DHCP	68	☐
use IP from EEPROM	--	☒

Figure 55: WBM Page “Port”

You are shown a list of all the protocols supported by the fieldbus controller.

5. Select the option **“DHCP”** or **“use IP from EEPROM”** to disable the BootP protocol.

You can also deactivate any other protocols that you no longer need in the same manner, or select desired protocols and activate them explicitly.

Since communication for each protocol takes place via different ports, you can have several protocols activated simultaneously; communication takes place via these protocols.

Note



The controller IP can be changed in the network by the DHCP server!

If DHCP is enabled and a DHCP server (e.g. a router with active DHCP server) is installed in the network, addresses are automatically allocated from the address range of the DHCP server after a power failure (failure of the 24 VDC power supply of the controller).

That is to say that all fieldbus coupler/controllers with active DHCP receive a new IP address!

Note



Check the network for a DHCP server when using DHCP!

If you are using DHCP, then make sure that there is a DHCP server in the network and that an IP address can be assigned to the fieldbus controller before start-up. BACnet is not initialized without an IP address and no application is launched.

6. Click on **SUBMIT** and then switch off the power to the fieldbus controller (hardware reset), or press down the mode selector switch. The protocol settings are then saved and the fieldbus controller is ready for operation.

If you have activated the MODBUS/TCP protocol, for example, you can now select and execute required MODBUS functions using the MODBUS master too, such as querying of the module configuration via register 0x2030.

If you have activated the WAGO-I/O-PRO for example, you can also program the controller via ETHERNET link using WAGO-I/O-PRO in line with Standard IEC-61131-3.

8.2.5.5 Reasons for Failed IP Address Assignment

- The PC on whom the BootP server is running is not located in the same subnet as the fieldbus controller; i.e., the IP addresses do not match.

Example:

Sub net mask: 255.255.255.0 (default value for fieldbus controllers)

PC IP: 192.168.2.100

Fieldbus controller IP: 192.168.1.200

Due to the sub net mask, the first 3 digits of the IP addresses must match.

- PC and/or controller is/are not linked to the ETHERNET
- Poor signal quality (use switches or hubs)

8.3 Testing the Function of the Fieldbus Node



Information

More information about reading the IP address

You can use WAGO-ETHERNET-Settings, button **[Identify]**, via Service Interface to read the IP addresses currently assigned.

1. To ensure that the IP address is correct and to test communication with the fieldbus node, first turn off the operating voltage of the fieldbus node.
2. Create a non-serial connection between your client PC and the fieldbus node.

After the power is switched on, the fieldbus controller is initialized. The fieldbus controller determines the I/O module configuration and creates a process image. During startup, the I/O LED (red) will flash. After a brief period, the I/O LED lights up green, indicating that the fieldbus controller is operational.

If an error occurs during start-up indicated by the I/O LED flashing red, evaluate the error code and argument and resolve the error.



Information

More information about LED signaling

The exact description for evaluating the LED signal displayed is available in the section "Diagnostics" > ... > "LED Signaling".

3. To test the coupler's newly assigned I/P address, start a DOS window by clicking on the **Start** menu item **Programs/MS-DOS Prompt**.
4. In the DOS window, enter the command: "**ping** " followed by the IP address of your coupler in the following format:

`ping [space] XXX . XXX . XXX . XXX (=IP address)`

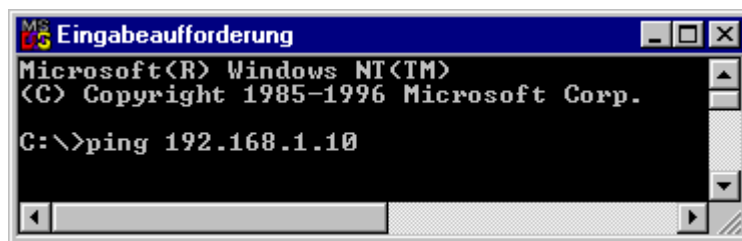


Figure 56: Example for the Function test of a Fieldbus Node

5. When the **[Enter]** key has been pressed, your PC will receive a query from the coupler, which will then be displayed in the DOS window. If the error message: "Timeout" appears, please compare your entries again to the allocated IP address and check all connections.

6. When the test has been performed successfully, you can close the DOS prompt.

The fieldbus node is now ready for communication.

8.4 Preparing the Flash File System

The flash file system must be prepared in order to use the Web interface of the fieldbus controller to make all configurations.

The flash file system is already prepared when delivered.

However, if the flash file system has not been initialized on your fieldbus controller or it has been destroyed due to an error, you must first extract it to the flash memory to access it.

NOTICE

Do not connect Communication Cable when energized!

To prevent damage to the communications interface, do not connect or disconnect 750-920 respectively 750-923 Communication Cable when energized! The fieldbus coupler/controller must be de-energized!

Note



Formatting erases data!

Note that formatting erases all data and configurations.

Only use this function when the flash file system has not been initialized yet or has been destroyed due to an error.

1. Switch off the supply voltage of the fieldbus controller.
2. Connect the communication cable 750-920 or 750-923 respectively the *Bluetooth*[®] Adapter 750-921 to the configuration interface of the fieldbus controller and to your computer.
3. Switch on the supply voltage of the fieldbus controller.

After the power is switched on, the fieldbus controller is initialized. The fieldbus controller determines the I/O module configuration and creates a process image. During startup, the I/O LED (red) will flash. After a brief period, the I/O LED lights up green, indicating that the fieldbus controller is operational.

If an error occurs during start-up indicated by the I/O LED flashing red, evaluate the error code and argument and resolve the error.

Information



More information about LED signaling

The exact description for evaluating the LED signal displayed is available in the section “Diagnostics” > ... > “LED Signaling”.

4. Start the **WAGO-ETHERNET-Settings** program.

5. In the top menu bar, select **Reset File System** to format the file system and to extract the Web pages of the flash file system.

Formatting and extracting is complete when the status window displays "Resetting the file system successfully".



Note

Restart the Fieldbus coupler/controller after resetting file system!

Make a restart of the fieldbus coupler/controller, so that the Web pages can be displayed after resetting file system.

8.5 Synchronizing the Real-Time Clock

The fieldbus controller's real-time clock enables a date and time indication for files in the flash file system.

At start-up, synchronize the real-time clock with the PC's current time.

There are two options to synchronize the real-time clock:

- Synchronize the real-time clock using **WAGO-ETHERNET-Settings**
- Synchronize the real-time clock using the **Web-based Management-System**

Synchronize the real-time clock using WAGO-ETHERNET-Settings

1. Switch off the supply voltage of the fieldbus controller.
2. Connect the communication cable 750-920 or 750-923 respectively the *Bluetooth*[®] Adapter 750-921 to the configuration interface of the fieldbus controller and to your computer.
3. Switch on the supply voltage of the fieldbus controller.

After the power is switched on, the fieldbus controller is initialized. The fieldbus controller determines the I/O module configuration and creates a process image. During startup, the I/O LED (red) will flash. After a brief period, the I/O LED lights up green, indicating that the fieldbus controller is operational.

If an error occurs during start-up indicated by the I/O LED flashing red, evaluate the error code and argument and resolve the error.



Information

More information about LED signaling

The exact description for evaluating the LED signal displayed is available in the section "Diagnostics" > ... > "LED Signaling".

4. Start the **WAGO Ethernet Settings** program.
5. Select the **Date and Time** tab.

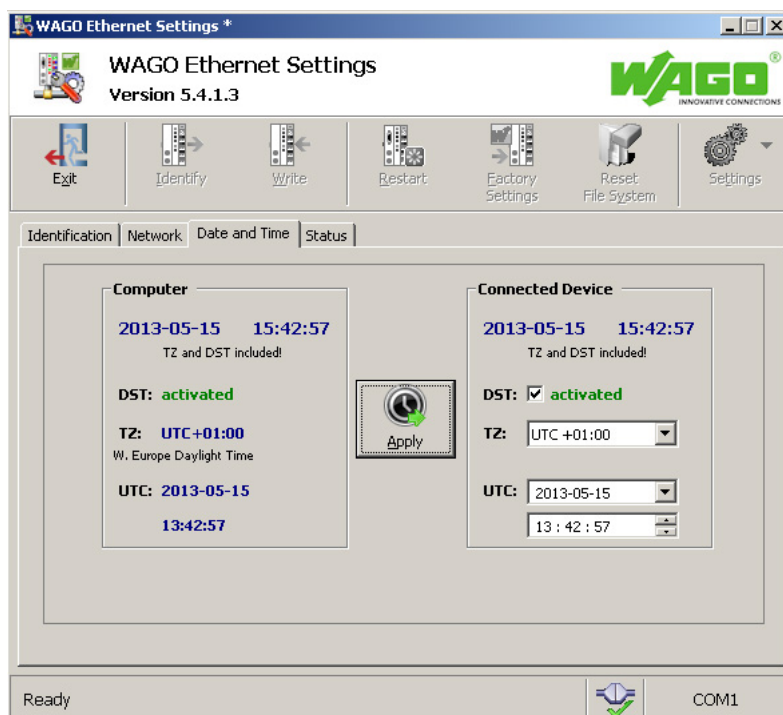


Figure 57: Example of Time Synchronization in ETHERNET Settings

6. Click on the **[Apply]** button.

Synchronize the real-time clock using the Web-based Management-System

1. Launch a Web browser (e.g., MS Internet Explorer or Mozilla) and enter the IP address you have assigned your fieldbus node in the address bar.
2. Click **[Enter]** to confirm.
The start page of the Web interface loads.
3. Select “Clock” in the left menu bar.
4. If an inquiry screen appears, enter your user name and password (default: user = “admin”, password = “wago” or user = “user”, password = “user”).
The HTML page “Clock configuration” loads.
5. Set the values in the fields “Time on device”, “Date” and “Timezone” to the current values and enable the desired “Daylight Saving Time (DST)” option.

Web-based Management



Clock configuration

Configuration Data

Time on device	11:07:54
Date (YYYY-MM-DD)	2015-11-19
Timezone (+/- hour:minute)	+1:00 ▾
Daylight Saving Time (DST)	☒ Automatic (USA) Second sunday in March 02:00 AM first sunday in November 02:00 AM
	☐ Automatic (EU) Last sunday in March 01:00 UTC last sunday in October 01:00 UTC
	☐ Manual: ☐
12 hour clock	☐

Figure 58: Example of WBM Clock Configuration

6. Click on **[SUBMIT]** to apply the changes in your fieldbus node.
7. Restart the fieldbus node to apply the settings of the Web interface.

8.6 Restoring Factory Settings



Note

For total deletion, first make a file system reset!

Note that you first reset the file system for the complete deletion of the controller. Use for this the program **WAGO-ETHERNET-Settings**, as described below, and click first on the button **[Reset File System]**. Then, you restore the factory settings.

To restore the factory settings, proceed as follows:

1. Switch off the supply voltage of the fieldbus controller.
2. Connect the communication cable 750-920 or 750-923 respectively the *Bluetooth*[®] Adapter 750-921 to the configuration interface of the fieldbus controller and to your computer.
3. Switch on the supply voltage of the fieldbus controller.
4. Start the **WAGO-ETHERNET-Settings** program.
5. In the top menu bar, select **[Factory Settings]** and click **[Yes]** to confirm.

A restart of the fieldbus node is implemented automatically. The start takes place with the default settings.

9 Programming the PFC Using WAGO-I/O-PRO

Using IEC 61131-3 programming, the BACnet/IP Controller 750-831 can also utilize the function of a PLC in addition to the functions of a fieldbus coupler. Creation of an application program in line with IEC 61131-3 is performed using the programming tool WAGO-I/O-PRO.

Information



Compatibility with IEC-61131-3 programming software!

The compatibility between your fieldbus controller and the IEC 61131-3 programming software used depends on the fieldbus controller's firmware version and the version of the programming software.

A list of recommended combinations can be found at www.wago.com.

Use the search function (search for “Compatibility Notes”).

Note



Activate option “CODESYS” in the web-based Management System for programming!

Pay attention, the IEC 61131-3 programming of the controller via ETHERNET requires that the check box **CODESYS** be activated at the Website “Port Configuration” (default).

You can, however, also connect the client PC and controller serially for programming using a programming cable.

Note



Log in with the administrator password to program the fieldbus controller!

If you have activated password protection for port 2455 on the “Security” page of the WBM, you have to log into WAGO-I/O-PRO in the menu **Online** > **Log In** to obtain programming access to the fieldbus controller (default password “wago”).



Figure 59: Logging in for Programming Access

Note



Log in to use the PLC browser with administrator user data!

To use the PLC browser functionality in WAGO-I/O-PRO, log into the PLC browser with the administrator user data (default user “admin”, password “wago”). Enter “login admin wago” in the command line of the PLC browser.

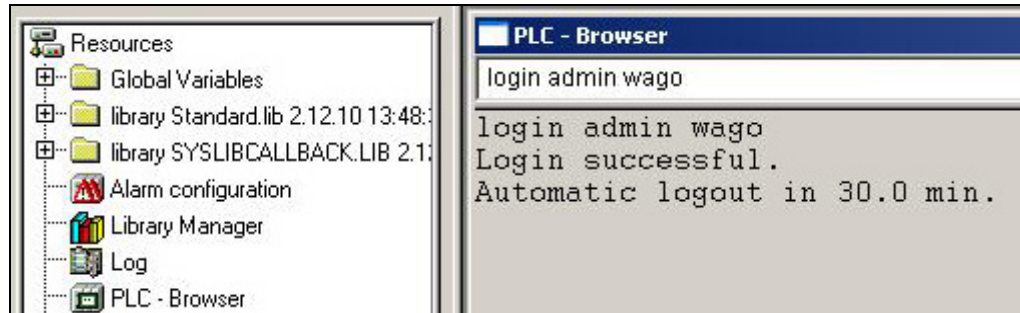


Figure 60: Logging Into the PLC Browser

A description of programming using WAGO-I/O-PRO is not included in this manual. The following sections, on the other hand, contain important information about creating projects in WAGO-I/O-PRO and about special modules that you can use explicitly for programming of the BACnet/IP Controller.

Explanations are also provided as to how the IEC 61131-3 program is transferred and how suitable communication drivers are loaded.

Note



One WAGO-I/O-PRO/(CODESYS)-Instance per target system!

Note that a simultaneous connection of multiple WAGO-I/O-PRO/(CODESYS) Instances on one target system is not possible.

Note



Name Conventions for WAGO-I/O-PRO/(CODESYS) Projects!

Note that you do not use special characters for the name of your WAGO-I/O-PRO/(CODESYS) project and limit the name to a maximum of 8 characters.

This will ensure that not always, in case of the online change function is activated simultaneously, for each online change event a new TxT file is created, which contains the paths and the project ID, and that for this additional memory is consumed. With proper choice of the file name, the TxT file is only overwritten each time and does not consume additional memory space.



Information

Additional Information:

For a detailed description of using the software, refer to the manual for the “WAGO-I/O-PRO”. This manual is located in the Internet under <http://www.wago.com>.

1. Start the programming tool at **Start \ Programs \ WAGO-I/O-PRO**.
2. Under **File / New** create a new project

A dialog window then appears on which you can set the target system for programming.

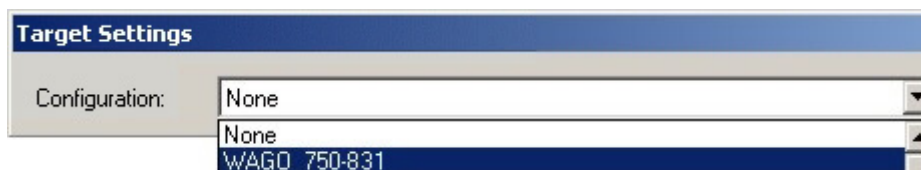


Figure 61: Dialog Window for Target System Settings

3. Select the 750-831 by entering **WAGO_750-831** and then click **[OK]**.
4. In the dialog window that appears select the program type (AWL, KOP, FUP, AS, ST or CFC).

To ensure that you can access all I/O module data properly in your new project, first compile the I/O module configuration based on the existing fieldbus node hardware and map it in the configuration file “EA-config.xml”.

This file defines whether write access is permitted to the modules from the IEC 61131-3 program, from the MODBUS/TCP or from BACnet.

As described below, this file can be generated via configuration using the WAGO I/O Configurator.

9.1 Configuring the Fieldbus Controller using the I/O Configurator

The I/O Configurator is a plug-in integrated into WAGO-I/O-PRO used to determine addresses for I/O modules at a fieldbus controller.

1. In the left half of the screen for the WAGO-I/O-PRO interface, select the tab **Resources**.
2. To start the I/O Configurator, double-click in the tree structure on **Control system configuration**.
3. Expand the branch **Hardware configuration** in the tree structure.
4. Right-click on the entry **K-Bus** and then select **Edit** in the context menu.
5. In the “Configuration” window that then opens, click on **Add** to open the module selection window.
6. Select the I/O module you wish to add from the module catalog and attach it to the end of the internal data bus structure by clicking on [**>>**] and **OK**.
7. Position all of the required I/O modules in their correct order until this arrangement matches the configuration of the physical node.

Arrange the tree structure in the hardware configuration in the same manner. Include all I/O modules which supply or receive data.

If you access your fieldbus controller online, you can use the [**Start WAGO-I/O-CHECK and scan**] button in the “Configuration” window to read in the physically linked fieldbus controllers with the series-connected I/O modules and display all of the components.

Note



The internal data bus structure in the WAGO I/O Configurator must match the physical node structure!

The number of I/O modules that send or receive data must correspond to the existing hardware (except for supply modules, copying modules or end modules, for example). For the number of input and output bits or bytes of the individual I/O modules please refer to their corresponding descriptions.



Information

Additional information

To open the data sheet for an I/O module, click in the “Configuration” window on the corresponding I/O module and then click the **[Data sheet]** button. The data sheet is then shown in a separate window.

All current data sheets are available on our website <http://www.wago.com> under Documentation.

8. Click **[OK]** to accept the node configuration and close the dialog window.

The addresses for the control system configuration are then recalculated and the tree structure for the configuration is updated.

If required, you can also modify the authorization privileges for individual I/O modules if they are to be accessed via fieldbus (MODBUS/TCP/IP). Initially, write access from the PLC is defined for each module that is added. Proceed as follows to change this setting:

9. Click the “PI Assignment” tab in the “Configuration” dialog.
10. Define for each module from where access to the module data is to be carried out.

You can choose from the following settings in the column “PI Assignment” for this:

- BACnet, fieldbus3 - Access from BACnet/IP
- MODBUS TCP/UDP, fieldbus1 - Access from MODBUS/TCP
- PLC, PLC (standard setting) - Access from PLC

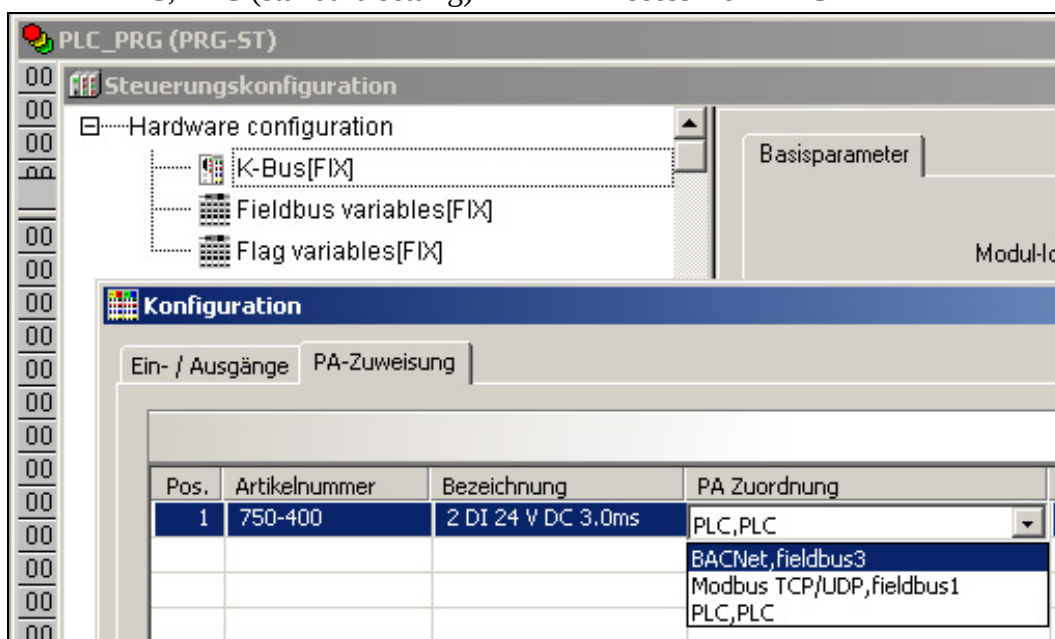


Figure 62: Write Access Over Module Parameters

After completing these settings you can begin the IEC-61131-3 programming.

An “EA-config.xml” configuration file is automatically generated and stored in the fieldbus controller, when you transfer the project (Menu **project > transfer/transfer all**) and download it in the fieldbus controller.

Note



Set “MODBUS TCP/UDP, fieldbus1”, when directly writing to a hardware address via MODBUS!

Set “MODBUS TCP/UDP, fieldbus 1” if you wish to write directly to a hardware address via MODBUS. Otherwise the modules will be allocated to the PLC, making writing from a different location impossible.

Information



Additional Information

For a detailed description of using the software WAGO-I/O-PRO and the I/O Configurator, refer to the online Help function for WAGO-I/O-PRO.

9.1.1 Configuration using the “EA-config.xml” File

You can also create the file “EA-config.xml” using an editor and store it in the controller directory “/etc” by means of FTP.

Configuration using the file “EA-config.xml” that is already stored in the fieldbus controller is described in this section.

Note



Configuration entries in WAGO-I/O-PRO overwrite “EA-config.xml” upon download!

If you wish to perform module assignment directly using the “EAconfig.xml” file stored in the controller, do not save any configuration data in WAGO-I/O-PRO prior to this, as the file is overwritten by entries in the WAGO-I/O-PRO on each download.

1. Open any FTP client. You can also use the Windows FTP client in the DOS prompt window:

[ftp://\[IP address of controller\]](ftp://[IP address of controller]), e.g. <ftp://192.168.1.201>

2. Then, enter **admin** as the user login and **wago** as the password.

The file “EA-config.xml” is located in the “etc” folder.

3. Copy this file to a local directory on your PC and open it in an editor installed on your PC (e.g., “WordPad”).

The file already contains the following syntax:

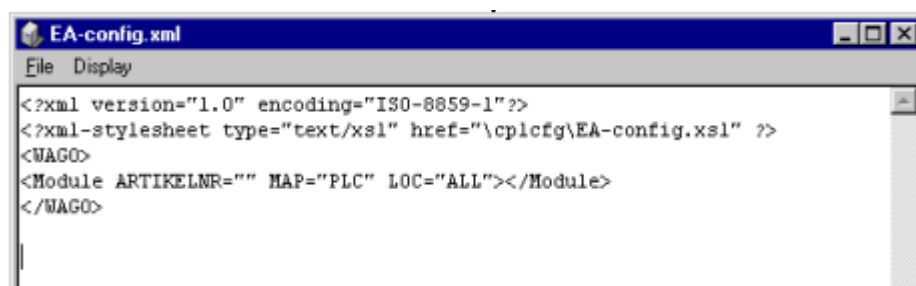


Figure 63: EA-config.xml

The fourth line contains the necessary information for the first I/O module. The entry MAP=PLC assigns write access privileges to the IEC-61131-3 program for the first module.

4. If you wish to change the access rights, replace "PLC" with "FB1" as the access privileges from MODBUS TCP/UDP.
If you wish to change the access rights, replace "PLC" with "FB2" as the access privileges from MODBUS RTU.
If you wish to change the access rights, replace "PLC" with "FB3" as the access privileges from BACnet:

```
<Module ARTIKELNR=" " MAP="PLC" LOC="ALL"> </Module>
```



```
<Module ARTIKELNR=" " MAP="FB3" LOC="ALL"> </Module>
```

5. Then complete the fourth line for each individual module using this syntax and set the corresponding assigned access privileges.

Note



The number of line entries must correspond with the number of bus terminals used!

It is imperative that the number of line entries concurs with the number of existing hardware modules.

6. Save the file and reload it to the controller file system via FTP client.

You can then begin with IEC-61131-3 programming.

Information



Additional Information:

For a detailed description of how to use the software, refer to the WAGO-I/O-PRO manual. The manual is available in the Internet under:

<http://www.wago.com> → Documentation → WAGO Software → WAGO-I/O-PRO → 759-333

9.2 Automatically prepare SYM_XML file

The "SYMXML.SYM_XML" file (called SYM_XML file in the following) is a file in the "xml" format, which is used to provide application variables of a WAGO-I/O-PRO program as BACnet objects in the BACnet Configurator.

The SYM_XML file is automatically created when downloading a WAGO-I/O-PRO project on the fieldbus controller and is available in the BACnet configurator without an additional import, if you perform the following steps:

1. In the WAGO-I/O-PRO software under **Project \ Options**, select category **Symbol configuration**.
2. Click on the **[configure symbol file...]** button to open the "set object attributes" window.
3. Select all elements in the top window area.

The SYM_XML file is written to the flash memory of the fieldbus controller. In order to keep the file size as small as possible, the required elements are filtered and only these are stored in the SYM_XML file.

4. Place a checkmark in the check box **Issue object variables**.

Once these options have been activated, an SYM_XML file will be generated automatically with project variables when a project is compiled.

5. Place a checkmark in the check box **export collective entries**, so that the objects of the unit are also created in SYM_XML.

Note



If possible, only enable the top two check boxes!

If possible, only enable the top two check boxes **export variables of the object** and **export collective entries**.

When selecting additional options, a download may not be made on the BACnet/IP controller, since the BACnet configurator does not filter the required files.

-
6. Confirm your entry with **[OK]**.
 7. Create your WAGO-I/O PRO project.

Note



Log in with the administrator password to program the fieldbus controller!

If you have activated password protection for port 2455 on the “Security” page of the WBM, you have to log into WAGO-I/O-PRO in the menu **Online** > **Log In** to obtain programming access to the fieldbus controller (default password “wago”).



Figure 64: Logging in for Programming Access

8. In the menu bar, click on **Online** and **Log-in**, to download the program on the fieldbus controller.

A window provides information of the changed program and requests, if the program should be downloaded.

9. Confirm with **[Yes]**.

Another window queries whether the BACnet objects used must be updated.

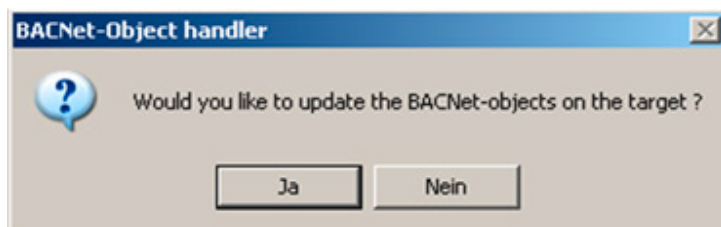


Figure 65: Updating the BACnet Image

10. Confirm with **[Yes]**.

The update of the configuration file is started. A loading icon is displayed. Meanwhile, the following actions are performed:

- The **Override file** is uploaded to the PC by the fieldbus controller and updated with the current information of the just created SYM_XML file. The updated override file is again downloaded on the fieldbus controller.
- The created **SYM_XML file** is updated and filtered for unnecessary entries. The file is downloaded on the fieldbus controller.
- The created **WAGO-I/O-PRO program** is downloaded.

11. To create a new boot project, first select the menu item **Create boot project**.
12. In order to restart the fieldbus controller, log off in WAGO-I/O-PRO via **Online > Log of**.

The SYM_XML and the override file are only reloaded by the fieldbus controller after this restart and changes are then included. The BACnet figure is adjusted. An import of the SYM_XML file is not required within the BACnet configurator.

Note



Check the configuration when the flash code 6-10 is displayed!

If flash code 6-10 is generated on the fieldbus controller, check if the control configuration corresponds with that of the node or if the SYM_XML file corresponds with the currently running WAGO-I/O-PRO program.

Note the change in the blink code (now 6 – 10) in comparison to BACnet/IP controller 750-830 (6 – 9).

9.3 Create SYM_XML file manually

Note



End the simulation mode for the symbol configuration!

If you are in the simulation mode you can not perform configuration of symbols or settings for generating the SYM_XML file. The category **Symbol configuration** is not available for selection in this case.

You can make this category visible by selecting **Online \ Logoff** in the main menu and removing the check mark in front of **Simulation**.

The SYM_XML file is a file that contains all project variables. This file is necessary for the BACnet Configurator.

Proceed as follows to create this file:

1. In the WAGO-I/O-PRO software under **Project \ Options** select the category **Symbol configuration**.
2. Place a check mark in the check box **Generate XML Symbol table**.
3. Click the button **Configure symbol file....** A dialog window then appears in which you can set the object attributes.
4. Place a check mark in the check box **Issue object variables**. The check marks must appear in black!

Note



Export the SYM_XML files only, if the set check mark is displayed in black!

If the check mark in the check box Issue object variables is illustrated in grey, it is not set for all minor elements. Click in the box again to make sure that the check mark appears in black.

Once these options have been activated, an SYM_XML file will be generated automatically with project variables when a project is compiled.

5. Place a check mark in the check box **export collective entries**, so that the objects of the unit are also created in the SYM_XML file.

The file is stored in the same directory as the WAGO-I/O-PRO program and can be imported in the BACnet configurator for further use.

9.4 ETHERNET Libraries for WAGO-I/O-PRO

Various libraries are available in WAGO-I/O-PRO for different IEC 61131-3 programming tasks. These libraries contain function blocks that can be used universally to facilitate and accelerate the creation of programs.

Once the libraries have been integrated, function blocks, functions and data types will be available that you can use the same as ones you have specifically defined.



Information

Additional Information

All libraries are included on the installation CD for the software WAGO-I/O-PRO or in the Internet under <http://www.wago.com>.

The libraries described below are specific to ETHERNET projects with WAGO-I/O-PRO.

Table 56: ETHERNET Libraries for WAGO-I/O-PRO

Library	Description
Ethernet.lib	Function blocks for communication via ETHERNET
WAGOLibEthernet_01.lib	Function blocks that can set up a link to a remote server or client via TCP protocol to exchange data with any potential UDP server or client via UDP protocol
WAGOLibModbus_IP_01.lib	Function blocks with MODBUS master function that set up links with one or more slaves to exchange data via UDP and TCP
SysLibSockets.lib	Function block for access to sockets for communication via TCP/IP and UDP
WagoLibSockets.lib	Function blocks for access to sockets for communication via TCP/IP and UDP Contains additional functions in addition to SysLibSockets.lib.
Mail_02.lib	Function block for sending e-mails
WAGOLibMail_02.lib	Function block for sending e-mails
WagoLibSnmpEx_01.lib	Function blocks for sending SNMP-V1 traps together with the parameters for the type DWORD and STRING(120) (starting with software version SW >= 07)
WagoLibSntp.lib	Function blocks for setting and using the simple network time protocol (SNTP)
WagoLibFtp.lib	Function blocks for setting and using the file transfer protocol (FTP)
WAGOLibTerminalDiag.lib	Function blocks for the output of module, channel and diagnostic data of I/O modules that provide diagnostic data
WAGOLibKBUS.lib	Function blocks for permitting both task-synchronous and consistent access to process data, as well as register communication between fieldbus coupler/controller and connected I/O modules



Information

Additional Information

For a detailed description of the function blocks and use of the software, refer to the online Help function for WAGO-I/O-PRO or the WAGO-I/O-PRO manual in the Internet under: <http://www.wago.com>.

9.5 Functional Restrictions and Limits

The basis of WAGO-I/O-PRO, the standard programming system CoDeSys by 3S, has an integrated visualization. Dependend on the target, this visualization can be used in the variants “HMI”, “TargetVisu” and “WebVisu”.

The fieldbus controller supports the process variants “HMI” and “WebVisu”. Depending on the version, there are technological limitations.

Several options for complex visualization objects “Alarm” and “Trend” are only provided by the “HMI” version. This applies, for example, to sending emails as a response to an alarm or for navigating through and generating historical trend data.

Compared with “HMI”, the “WebVisu” on the fieldbus controller is executed within considerably tighter physical limits. Whereas the “HMI” can call upon the resources of a PC, the “WebVisu” operate within the following restrictions:

File system (4,5 MB):

The overall size of the PLC program, visualization files, bitmaps, log files, configuration files, etc. must fit into the file system.

The PLC browser delivers the amount of free disk space in response to the command “fds” (FreeDiscSpace).

Process data buffer (16 kB):

The WebVisu uses its own protocol for exchanging process data between applet and control system. In doing so, the process data is transmitted with ASCII coding. The pipe character (“|”) separates two process values. For this reason, the required space of a process data variable in the process data buffer not only depends on the data type, but also on the process values itself. A “WORD” variable therefore occupies between one byte for the values 0...9 and five bytes for values greater than 10000. The selected format allows only a rough estimate of the space required for the individual process data in the process data buffer. If the size is exceeded, the WebVisu no longer works as expected.

The number of modules (1023/default):

The total size of the PLC program is determined, among other things, by the maximum number of modules. This value can be configured in the target system settings.

Computing power/processor time:

The 750-831 is based on a real-time operating system with pre-emptive multitasking. High-priority processes such as the PLC program will eliminate low-priority processes.

The web server supplies process data and applets for the web visualization. Make sure when configuring tasks, that there is sufficient processor time available for all processes. The “freewheeling” task call option is not suitable in conjunction with the “WebVisu”; as in this case, the high-priority PLC program suppresses the web server. Instead of this, use the “cyclic” task call option with a realistic value.

The PLC browser provides an overview of the real execution times for all CoDeSys tasks with the command “tsk”.

If in a PLC program, operating system functions are used; e.g., for the handling of “sockets” or the “file system”, these execution times are not taken into consideration covered by the command “tsk”.

CTU counter:

The CTU counter operates in a value range of 0 to 32767.

Note



Note the maximum number of write cycles of the EEPROM!

Fieldbus couplers/controllers save some information such as IP addresses and IP parameters in the EEPROM to make it available after a restart. The memory cycles of an EEPROM are generally limited. Beyond a limit of approx. 1 million write cycles, memory can no longer be assured. A defective EEPROM only becomes apparent after a restart by software reset or power-on.

Due to a bad checksum, the fieldbus coupler/controller then always starts with the default parameters.

The following functions use the EEPROM:

- **WAGO-I/O-PRO**
 - **WagoLibDaylightSaving** SetDaylightSavings
 - **EthernetLib** SetNetworkConfig
SetVariables
- **MODBUS**
 - Register 0x1035 Time Offset
 - Register 0x100B Watchdog parameters
 - Register 0x1028 Network configuration
 - Register 0x1036 Daylight saving
 - Register 0x1037 Modbus response delay
 - Register 0x2035 PI parameter
 - Register 0x2043 Default configuration
- **Parameter assignments**
 - **BootP** new parameters
 - **DHCP** new parameters
 - **WAGO MIB** write access

Note



Limited functions when RAM used to full capacity!

Note that some limitations (e.g., reaction time) may apply when the RAM is used to full capacity.

9.6 General Information about IEC Tasks

Please note the following information when programming your IEC tasks:



Note

Use different priorities for IEC tasks!

Give IEC tasks (in WAGO-I/O-PRO under the **Resources** tab > **Task Configuration**) different priorities to prevent an error from occurring when compiling the user program.

An interruption of IEC tasks is possible through tasks of higher priority!

An ongoing task may be interrupted by tasks with higher priorities. Execution of the task that has been interrupted is resumed only when there are no other higher-priority tasks to be executed.

Distortion of variables in overlapping areas of the process image!

If several IEC tasks utilize input or output variables with the same, or overlapping addresses in the process image, the values for the input or output variables may change while the IEC task is being executed!

Observe waiting periods of free-running tasks!

Running tasks are halted after each task cycle for half the time that the task proper requires (min. 1 ms). Execution of the task is then resumed.

Example: 1st Task 4 ms → Waiting period 2 ms
 2nd Task 2 ms → Waiting period 1 ms

The default task is created by default!

If no task has been defined in the task configuration, a running default task is created during translation. This task, called “Default task”, is recognized by this name in the firmware, meaning that the name “Default task” cannot be used for other task names.

Observe the watchdog sensitivity for cyclic tasks!

The watchdog sensitivity indicates how many times the watchdog time is exceeded for an even to be triggered. You set the sensitivity in WAGO-I/O-PRO under Register **Resources** > **Task Configuration** for Cyclical Tasks. The values 1 and 0 are equivalent with regard to sensitivity. A sensitivity value of 0 or 1 results in the watchdog event being triggered when the watchdog time is exceeded on time. With a sensitivity value of 2, for instance, the watchdog time must be exceeded in two consecutive task cycles in order for the watchdog event to be triggered.

The following applies to cyclic tasks with watchdog activated:

Note



Reference for Watchdog Settings!

For each tasks created, a watchdog can be enabled that monitors the execution time of a task.

If the task runtime exceeds the specified watchdog time (e.g., t#200 ms), then the watchdog event has occurred.

The runtime system stops the IEC program and reports an error.

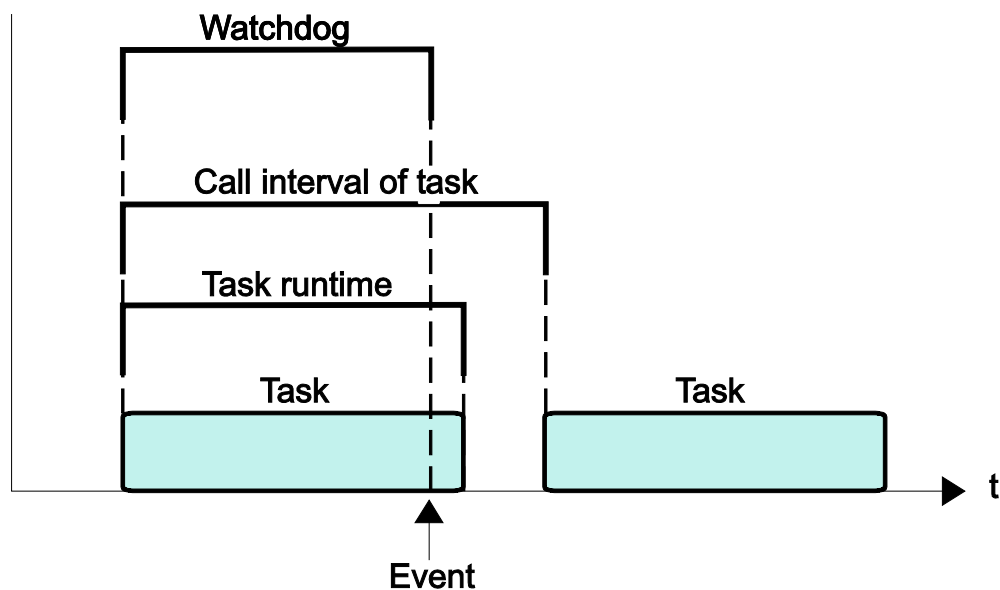


Figure 66: Watchdog Runtime is Less Than the Task Runtime

If the watchdog time set is greater than the call interval of the task, then the watchdog is restarted for each task call interval.

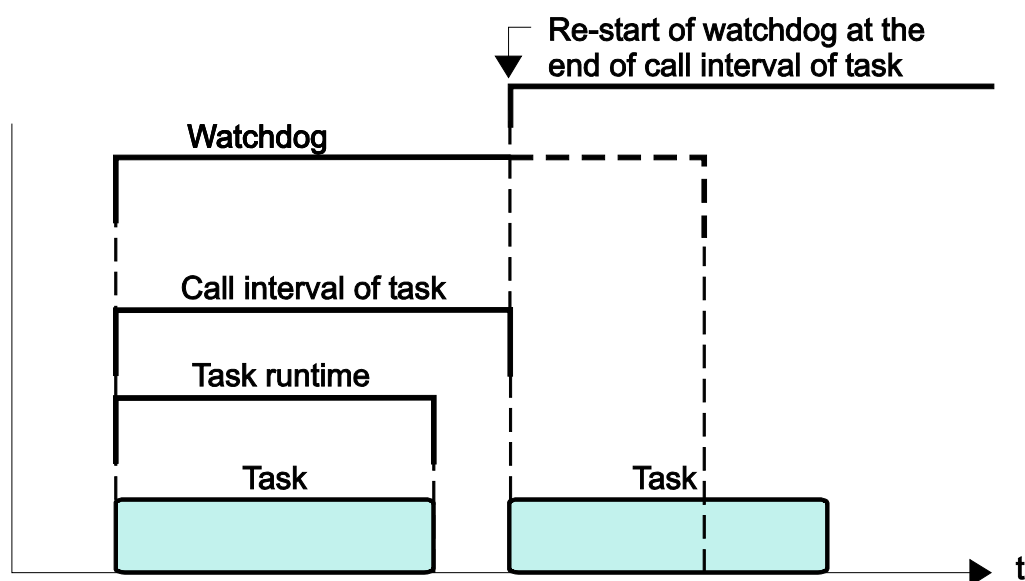


Figure 67: Watchdog Runtime Is Greater Than the Task Call Interval

9.6.1 IEC Task Sequence

1. Determine the system time (tStart).
2. If no full internal bus cycle has run since the last time the outputs were written:
→ Wait until the next internal bus cycle is completed.
3. Reading of inputs and reading back of the outputs from the process image.
4. If the application program has been started.
→ Execute the program codes for this task.
5. Writing of the outputs to the process image.
6. Determine the system time (tEnd).
→ $tEnd - tStart = \text{runtime for the IEC task}$

9.6.2 Overview of Most Important Task Priorities

Table 57: Task Processing

Task	Importance of the execution
Internal bus task, fieldbus task	of priority before all others
Normal task	after the internal bus and fieldbus tasks
PLC-Comm task	after the normal tasks
Background task	after the PLC-Comm tasks

Internal bus task/fieldbus task (internal)

The internal bus task matches the process image to the input/output data of the modules in defined cycles. The fieldbus tasks are performed as triggered by events and only require computing time when communication is performed via fieldbus (MODBUS).

Normal task (IEC-Task priorities 1-10 that can be set in CODESYS)

IEC tasks with this priority may be interrupted by the internal bus tasks. Therefore, configuration for the connected modules and communication via fieldbus with the watchdog activated for the task call interval must be taken into account here.

PLC-Comm task (internal)

The PLC-Comm task is active when logged in and takes up communication with the CODESYS gateway.

Background task (IEC-Task priorities 11-31 that can be set in CODESYS)

All internal tasks have a priority higher than that for the IEC background tasks. These tasks are therefore very well-suited for performing time-intensive and non-critical time tasks, such as calling up functions in the SysLibFile.lib.



Information

Additional Information

For a detailed description of using the software, refer to the manual for the “WAGO-I/O-PRO”. This manual is located in the Internet under <http://www.wago.com>.

9.7 System Events

In place of a task, a system event can also call up a project module for processing.

The system events to be employed for this depend on the target system. These events consist of the list of supported standard system events for the control system and any other manufacturer-specific events which may have been added.

Possible events, for example: Stop, Start, Online change.

A complete list of all system events is provided at WAGO-I/O-PRO in tab **Resources > Task configuration > System events**.

9.7.1 Enabling/Disabling System Events

1. Open the register **resources > task configuration > system events** in WAGO-I/O-PRO (see the following Figure).
2. In order to call up a module via an event, activate the entries by setting a hatch mark in the respective control boxes.
3. Disable the control boxes by removing the hatch marks through a mouse click.

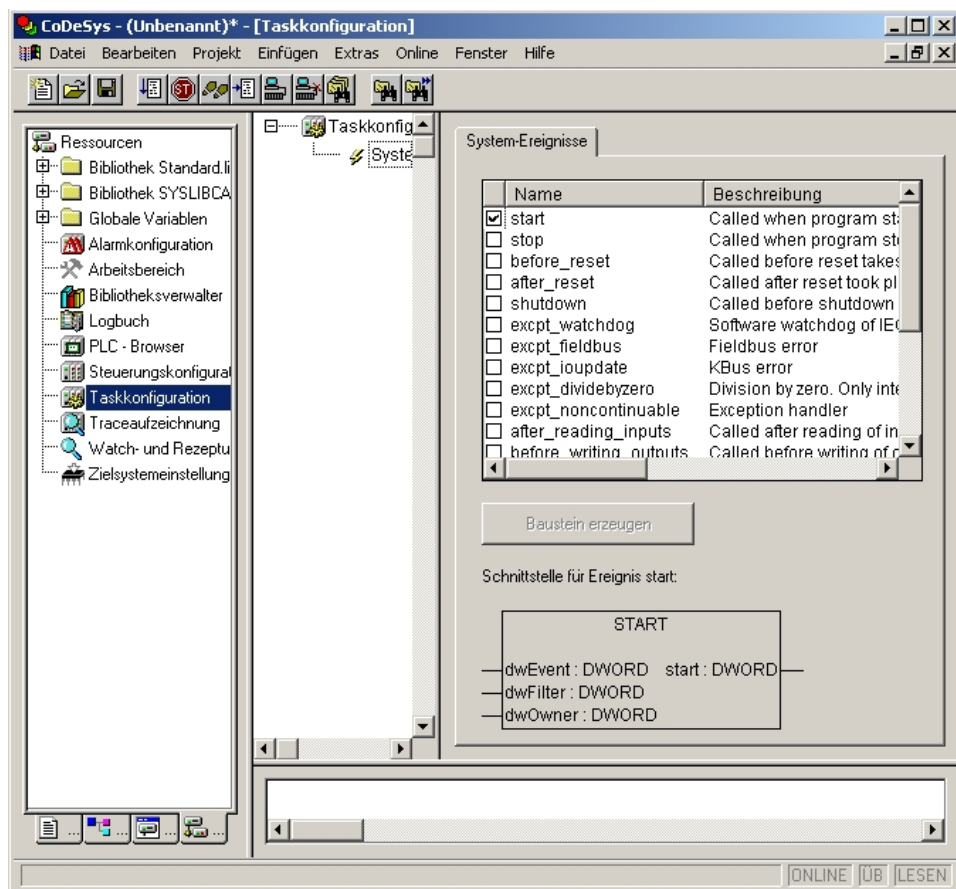


Figure 68: Enabling/Disabling System Events

Note



Only use global variables and functions (Fun)!

Note that the fetch of function blocks (FB) such as Timer modules , or the fetch of blocking functions such as file operations can lead to a malfunction of the system. Therefore use in system events exclusively Global variables and functions (Fun).

Information



Additional Information:

Allocation of the system events to the specific modules to be called up is clarified in the manual for the programming tool WAGO-I/O-PRO in the Internet under <http://www.wago.com>.

9.8 Transfer the IEC program to the Fieldbus controller

Transfer from the PC to the fieldbus controller of the program for the created IEC-61131-3 application can be performed different ways, e.g. (see following sections):

- Direct transfer via serial port
- Transfer via fieldbus

Suitable communication drivers are required for transfer; these can be loaded and configured using WAGO-I/O-PRO.

Note



Check/adjust communications parameters of the driver

When selecting the desired driver, watch for the proper settings and adjustments of the communications parameters (see the following description).

Note



Transmission only on matching target!

Note that a boot project can generally be transferred only to fieldbus controllers whose item-number has been set as target during compilation.

Note



“Reset” and “Start” are required to set the physical outputs!

The initialization values for the physical outputs are not set immediately after downloading. Select **Online > Reset** and subsequently **Online > Start** in the menu bar of WAGO I/O-PRO to set the values.

Note



Stop application before generating large boot projects!

Stop the WAGO-I/O-PRO application via **Online > Stop** before generating a very large boot project, since this may otherwise cause stopping the internal bus. You can restart the application after creating the boot project.

Note



Handling persistent data affects the program start!

Depending on the variable type, the number and sizes of the persistent data and their combination, such as in function modules, handling with persistent data can delay the program start by an extended initialization phase.



Information

Additional Information

The following description is used for fast access. For details on installing missing communication drivers and using the software, refer to “WAGO-I/O-PRO” available in the Internet under <http://www.wago.com>.

9.8.1 Transfer via Serial Service Port



Note

Watch the position of the mode selector switch when accessing the fieldbus controller!

Prerequisite for the access to the fieldbus controller is that the operating mode switch of the fieldbus controller, which is located behind the cover of the fieldbus controller next to the service interface, is in the center or top position.

Use the WAGO communication cable to set up a physical connection via serial service port. This cable is included in the scope of supply for the IEC-61131-3 programming tool (order no. 759-333), or can be procured as an accessory item under order no. 750-920.

NOTICE

Do not connect Communication Cable when energized!

To prevent damage to the communications interface, do not connect or disconnect 750-920 respectively 750-923 Communication Cable when energized! The fieldbus coupler/controller must be de-energized!

1. Check that the fieldbus controller mode selector switch is set to the center or top position.
If this is not the case, move the mode selector switch to the center or top position.
2. Use the WAGO communication cable to connect a COM port of your PC to the fieldbus controller communication port.

A communication driver is required for serial data transfer. This driver and its parameters must be entered in the WAGO-I/O-PRO in the dialog window "Communication parameters".

3. Start the WAGO-I/O-PRO software **under Start > Programs > WAGO Software > WAGO-I/O-PRO.**
4. In the menu **Online** select the item **Communication parameters.**

The dialog window "Communication parameters" then appears. The channels of the currently connected gateway servers are shown on the left side of the dialogue and the already installed communications drivers are shown below. This window is empty in its default settings.

5. Click **New** to set up a link and then enter a name, such as RS-232 Connection.

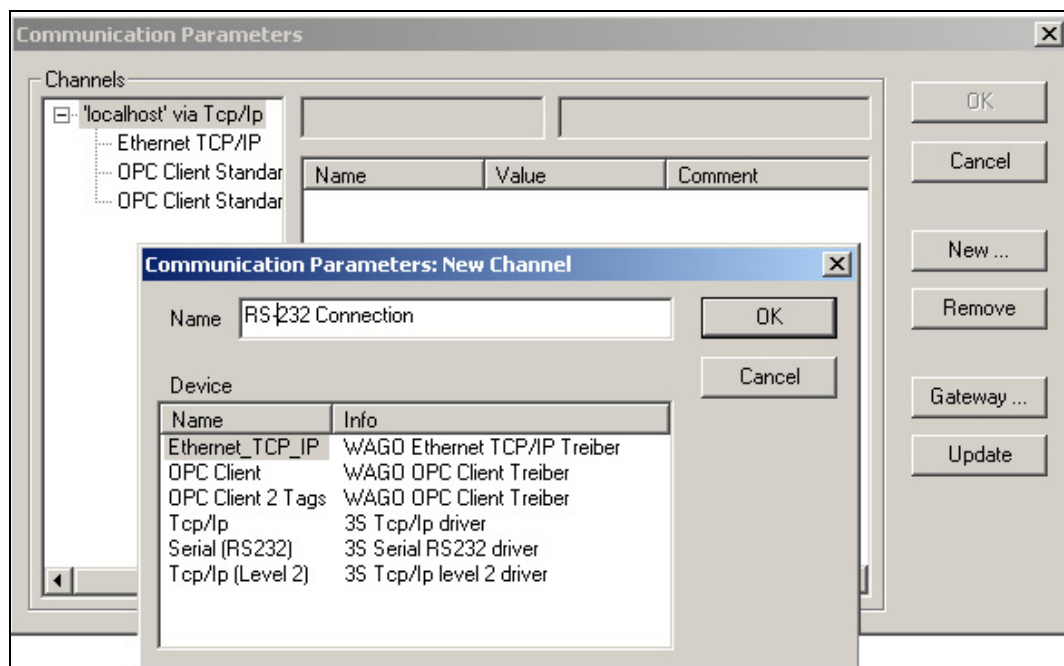


Figure 69: Dialog Window "Communication Parameters"

6. In the selection window, mark the required driver in the right side of the window, Serial (RS-232) 3S Serial RS-232 driver, to configure the serial link between the PC and the fieldbus controller.

The following properties for the serial port are shown in the center dialog window:

- Port: COM1
- Baud rate: 19200
- Parity: Even
- Stop-bits: 1
- Motorola byte order: No

7. If necessary, change the entries according to the above values by clicking on the respective value and editing it.
8. Confirm these settings by clicking **OK**

The RS-232 port is now configured for transferring the application.

Note



Log in with the administrator password to program the fieldbus controller!

If you have activated password protection for port 2455 on the “Security” page of the WBM, you have to log into WAGO-I/O-PRO in the menu **Online** > **Log In** to obtain programming access to the fieldbus controller (default password “wago”).



Figure 70: Logging in for Programming Access

9. Under **Online**, click the menu item **Login** to log in to the fieldbus controller

The WAGO-I/O-PRO Server is active during online operation. The communication parameters can not be called up during this time.

Depending on whether a program is already present in the fieldbus controller, a window will appear asking whether a (new) program should be loaded.

10. Respond with **Yes** to load the current program.
11. In menu **Online**, click on **Create Boot project**.

Your compiled project will also be executed by this method, if you restart the fieldbus controller or if there is a power failure.

12. Once the program has been loaded, start program processing in the menu **Online**, menu item **Start**.

This command starts the processing of your program in the control system or in the simulation.

“ONLINE” and “RUNNING” will then appear at the right of the status bar.

13. To terminate online operation, click the menu item **Log off** in the menu **Online**.

9.8.2 Transfer via ETHERNET

The physical link between the PC and the controller is set up via fieldbus. An appropriate communication driver is required for data transfer. The driver and its parameters must be entered in the WAGO-I/O-PRO in the dialog window "Communication parameters".



Note

Controller needs IP address for access!

The controller must have an IP address before it can be accessed. The operating mode switch, which is located behind the cover of the fieldbus controller next to the service interface, must be in the center or top position.

1. Start the WAGO-I/O-PRO software under **Start / Programs / WAGO-Software > WAGO-I/O-PRO**.
2. In the menu **Online** select the item **Communication parameters**.

The dialog window "Communication parameters" then appears. The channels of the currently connected gateway servers are shown on the left side of the dialogue and the already installed communications drivers are shown below. This window is empty in its default settings.

3. Click **New** to set up a connection and then specify a name, e.g. TcpIp connection.
4. Mark the required TCP/IP driver in the right side of the dialog window to configure the link between the PC and the controller via ETHERNET. Select the new driver version "**Tcp/Ip**" (3S Tcp/Ip driver).

The following standard entries are shown in the center dialog window:

- IP address: IP address of your controller
- Port number: 2455
- Motorolabyteorder: No

5. Change any entries as you may require.
6. Confirm with **OK**.

You have now configured the TCP/IP link with the communication parameters/drivers.

7. Under **Online**, click the menu item **Login** to log in to the fieldbus controller

The WAGO-I/O-PRO Server is active during online operation. The communication parameters can not be called up during this time.

Depending on whether a program is already present in the fieldbus controller, a window will appear asking whether a (new) program should be loaded.

8. Respond with **Yes** to load the current program.
9. In menu **Online**, click on **Create Boot project**.

You compiled project will also be executed by this method, if you restart the fieldbus controller or if there is a power failure.

10. Once the program has been loaded, start program processing in the menu **Online**, menu item **Start**.

This command starts the processing of your program in the control system or in the simulation.

“ONLINE” and “RUNNING” will then appear at the right of the status bar.

11. To terminate online operation, click the menu item **Log off** in the menu **Online**.

10 Configuring via the Web-Based Management System (WBM)

An internal file system and an integrated Web server can be used for configuration and administration of the system. Together, they are referred to as the Web-based Management System (WBM).

The HTML pages saved internally provide you with information about the configuration and status of the fieldbus node. In addition, you can also change the configuration of the device here.

You can also save HTML pages created yourself via the implemented file system.

Note



Always restart after making changes to the configuration!

The system must always be restarted for the changed configuration settings to take effect.

1. To open the WBM, launch a Web browser (e.g., Microsoft Internet Explorer or Mozilla Firefox).
2. Enter the IP address of the fieldbus coupler/controller in the address bar.
3. Click **[Enter]** to confirm.
The start page of WBM loads.
4. Select the link to the desired HTML page in the left navigation bar.
A query dialog appears.
5. Enter your user name and password in the query dialog (default: user = “admin”, password = “wago” or user = “user”, password = “user”).
The corresponding HTML page is loaded.
6. Make the desired settings.
7. Press **[SUBMIT]** to confirm your changes or press **[UNDO]** to discard the changes.
8. Restart the system to apply the settings (HTML page “Security”, button **[Software Reset]**).

You can access the following WBM pages via the links given in the navigation bar:

- Information
- Ethernet
- TCP/IP
- Port
- SNMP
- SNMP V3
- Watchdog
- Clock
- Security
- Modbus
- BACnet
- PLC Info
- PLC
- Features
- I/O config
- Disk Info
- SD Card
- Backup & Restore
- WebVisu

10.1 Information

The WBM page “Information” contains an overview of all important information about your fieldbus coupler/controller.

Web-based Management



Status information

Coupler details

Order number	750-xxx
Mac address	0030DE07B953
Firmware revision	01.01.07 (05)

Actual network settings

IP address	192.168.1.10
	Static Configuration
Subnet mask	255.255.0.0
Gateway	0.0.0.0
Host Name	0030DE07B953
Domain Name	
(S)NTP-Server	0.0.0.0
DNS-Server 1	0.0.0.0
DNS-Server 2	0.0.0.0

Module status

State Modbus Watchdog:	Disabled
Error code:	0
Error argument:	0
Error description:	Coupler running, OK

Figure 71: WBM Page "Information"

Table 58: WBM Page “Information”

Coupler details			
Entry	Default	Value (example)	Description
Order number	750-831	750-831	Order number
Mac address	0030DEXXXXXX	0030DE000006	Hardware MAC address
Firmware revision	kk.ff.bb (rr)	01.01.09 (00)	Firmware revision number (kk = compatibility, ff = functionality, bb = bugfix, rr = revision)
Actual network settings			
Entry	Default	Value (example)	Description
IP address	0.0.0.0	192.168.1.80	IP address, Type of IP address assignment
Subnet mask	255.255.255.0	255.255.255.240	Subnet mask
Gateway	0.0.0.0	192.168.1.251	Gateway
Host name	___	___	Host name (not assigned here)
Domain name	___	___	Domain name (not assigned here)
(S)NTP server	0.0.0.0	0.0.0.0	Address of (S)NTP server
DNS server 1	0.0.0.0	0.0.0.0	Address of first DNS server
DNS server 2	0.0.0.0	0.0.0.0	Address of second DNS server
Module status			
Entry	Default	Value (example)	Description
State Modbus Watchdog	Disabled	Disabled	Status of Modbus Watchdog
Error code	0	10	Error code
Error argument	0	5	Error argument
Error description	Coupler running, OK	Mismatch in CODESYS IO-configuration	Error description

10.2 Ethernet

Use the “Ethernet” HTML page to set the data transfer rate and bandwidth limit for each of the two switch ports for data transfer via Ethernet.

Web-based Management

Ethernet configuration

This page is for the configuration of the Ethernet Switch and Aging settings. The configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

Phy Configuration

Description	Port 1	Port 2
Enable Port	☒	☒
Enable Autonegotiation	☒	☒
10 MBit Half Duplex	☐	☐
10 MBit Full Duplex	☐	☐
100 MBit Half Duplex	☐	☐
100 MBit Full Duplex	☐	☐

MAC Address Filter

Enable ☐

Whitelist format: xx:yy:zz:aa:bb:cc

WAGO 750/767 devices ☐

MAC 1	00:00:00:00:00:00
MAC 2	00:00:00:00:00:00
MAC 3	00:00:00:00:00:00
MAC 4	00:00:00:00:00:00
MAC 5	00:00:00:00:00:00

Misc. Configuration

Description	Port 1	Port 2	internal Port
Input Limit Rate	No Limit	No Limit	No Limit
Output Limit Rate	No Limit	No Limit	No Limit
Fast Aging	☐		
BC protection	☐	☐	
Port Mirror	☐		
Sniffer Port	☐	☒	
Mirror Port	☒	☐	☐
Ethernet MTU	1500		

Figure 72: WBM Page “Ethernet”

Table 60: WBM Page "Ethernet"

Misc. Configuration				
Entry	Port			Description
	1	2	int.	
Input Limit Rate	No Limit ▼			The Input Limit Rate limits network traffic when receiving. The rate is indicated in megabits or kilobits per second. If the limit is exceeded, packets are lost.
Output Limit Rate	No Limit ▼			The Output Limit Rate limits network traffic when sending. The rate is indicated in megabits or kilobits per second. If the limit is exceeded, packets are lost.
Fast Aging	☐			☒ Enable "Fast Aging" "Fast Aging" ensures that the cache for the MAC addresses is cleared faster in the switch. This may be required if a redundancy system (e.g., using a Jet-Ring network or comparable technology) needs to be set up.
				☐ Disable "Fast Aging". The time to discard the cache entries is five minutes.
BC protection	☐	☐		☒ Broadcast Protection limits the number of broadcast telegrams per unit of time. If protection is on, the broadcast packets are limited at 100 Mbit to 8 packets per 10 ms and at 10 Mbit to 8 packets per 100 ms. If the limit is exceeded, packets are lost.
				☐ Broadcast Protection disabled.
Port Mirror	☐			☒ Enable port mirroring Port Mirroring is used for network diagnostics. Packets are mirrored from one port (mirror port) to another (sniffer port).
				☐ Disable port mirroring
Sniffer Port	☐	☒		Select the sniffer port the mirror port should be mirrored to.
Mirror Port	☒	☐	☐	Select the mirror port which should be mirrored to the sniffer port.
Ethernet MTU	1500			Maximum packet size of a protocol, which can be transferred without fragmentation ("Maximum Transmission Unit" - MTU)

Note



Set the MTU value for fragmentation only!

Only set the value for MTU, i.e., the maximum packet size between client and server, if you are using a tunnel protocol (e.g., 1452 for VPN) for ETHERNET communication and the packets must be fragmented.

Setting the value is independent of the transmission mode selected.

Note



Configure ETHERNET transmission mode correctly!

A fault configuration of the ETHERNET transmission mode may result in a lost connection, poor network performance or faulty performance of the fieldbus coupler/controller.

10.3 TCP/IP

You can configure network addressing and network identification on the “TCP/IP” HTML page.

Web-based Management

TCP/IP configuration

This page is for the configuration of the basic TCP/IP network parameters. The parameters are stored in an EEPROM and changes will take effect after the next software or hardware reset.

Note that these settings are used only if the DIP switch is set to zero and you have selected 'use IP from EEPROM' at 'Port' configuration page! Otherwise the settings from DIP switch will be used!

EEPROM Configuration Data

IP-Address	192.168.1.111
Subnet Mask	255.255.255.0
Gateway	0.0.0.0
Host Name	0030DE0AA31B
Domain Name	
DNS Server 1	0.0.0.0
DNS Server 2	0.0.0.0
Switch IP-Address	192.168.1
(S)NTP Server Host Name Or IP Address	0.0.0.0
SNTP port 123 enabled	☐
SNTP Update Time (sec, max. 65535)	3600
IP Fragment TTL (sec, max. 255)	60

Figure 73: WBM Page “TCP/IP”

Table 61: WBM Page "TCP/IP"

EEPROM Configuration Data			
Entry	Default	Value (example)	Description
IP address	0.0.0.0	192.168.1.200	Enter IP address
Subnet mask	255.255.255.0	255.255.255.0	Enter subnet mask
Gateway	0.0.0.0	0.0.0.0	Enter gateway
Host name	0030DEXXX XXX	0030DE026005	Enter host name
Domain name			Enter domain name
DNS Server1	0.0.0.0	0.0.0.0	Enter IP address of the first DNS server
DNS Server2	0.0.0.0	0.0.0.0	Enter optional IP address of the second DNS server
Switch IP-Address	192.168.1	192.168.5	Network address for the configuration of the IP address with DIP switch
(S)NTP Server Host Name Or IP Address	0.0.0.0	0.0.0.0 ptbtime1.ptb.de	Enter IP address or host name (max. 63 characters) of the (S)NTP server
SNTP port 123 enabled	☐	☒	Activate „Simple Network Time Protocol“
		☐	Deactivate „Simple Network Time Protocol“
SNTP Update Time (sec. max. 65535)	3600	0	Enter the delay after which the (S)NTP server requests the network time again
IP Fragment TTL (sec. max. 255)	60	60	Life of a packet (Time to Live)

10.4 Port

Use the “Port” HTML page to enable or disable services available via the IP protocol.

Web-based Management

Port configuration

This page is for the configuration of the network protocols. The configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

Port Settings

Protocol	Port	Enabled
FTP	21	☒
SNTP	123	☐
HTTP	80	☒
HTTPS	443	☐
SNMP	161, 162	☒
Modbus UDP	502	☒
Modbus TCP	502	☒
WAGO Services	6626	☒
CoDeSys	2455	☒
BootP	68	☐
DHCP	68	☐
use IP from EEPROM	--	☒

UNDO
SUBMIT

Figure 74: WBM Page “Port”

Table 62: WBM page “Port”

Port Settings		
Entry	Entry	Entry
FTP (Port 21)	Enabled ☒	☒ activating “File Transfer Protocol”
		☐ deactivating “File Transfer Protocol”
SNTP (Port 123)	Enabled ☐	☒ activating "Simple Network Time Protocol"
		☐ deactivating "Simple Network Time Protocol"
HTTP (Port 80)	Enabled ☒	☒ activating "Hypertext Transfer Protocol"
		☐ deactivating "Hypertext Transfer Protocol"

HTTPS (Port 443)	Enabled ☐	☒ activating "Hypertext Transfer Protocol over SSL-encrypted connection"
		☐ deactivating "Hypertext Transfer Protocol over SSL-encrypted connection"
	 Note Operate WebVisu via HTTPS! You can operate WebVisu via HTTPS protocol, provided HTTPS is activated and you use a WAGO-I/O-PRO version >= .42.	
SNMP (Port 161, 162)	Enabled ☒	☒ activating "Simple Network Management Protocol"
		☐ deactivating "Simple Network Management Protocol"
Ethernet IP (TCP-Port 44818, UDP-Port 2222)	Enabled ☐	☒ activating ETHERNET/IP protocol
		☐ deactivating ETHERNET/IP protocol
Modbus UDP (Port 502)	Enabled ☒	☒ activating MODBUS/UDP protocol
		☐ deactivating MODBUS/UDP protocol
Modbus TCP (Port 502)	Enabled ☒	☒ activating MODBUS/TCP protocol
		☐ deactivating MODBUS/TCP protocol
WAGO Services (Port 6626)	Enabled ☒	☒ activating WAGO services
		☐ deactivating WAGO services
CODESYS (Port 2455)	Enabled ☒	☒ activating WAGO-I/O-PRO (CODESYS)
		☐ deactivating WAGO-I/O-PRO (CODESYS)
BootP (Port 68)	Enabled ☒	☒ activating "Bootstrap Protocol"
		☐ deactivating "Bootstrap Protocol"
DHCP (Port 68)	Enabled ☐	☒ activating "Dynamic Host Configuration Protocol"
		☐ deactivating "Dynamic Host Configuration Protocol"
use IP from EEPROM	Enabled ☐	☒ activating use of IP address from EEPROM
		☐ deactivating use of IP address from EEPROM



Note

Alternative IP address assignment!

You can only select the DHCP, BootP and "use IP from EEPROM" settings as an alternative!

10.5 SNMP

On the HTML page “SNMP”, you can perform the settings for the Simple Network Management Protocol.

SNMP is a standard for device management within a TCP/IP network. The Simple Network Management Protocol (SNMP) is responsible for transporting the control data that allows the exchange of management information, the status and statistic data between individual network components and a management system.

The fieldbus coupler/controller supports SNMP in versions 1, 2c and 3.

The SNMP of the ETHERNET TCP/IP controller includes the general MIB according to RFC1213 (MIB II).

SNMP is processed via port 161. The port number for SNMP traps (agent messages) is 162.

Note

**Enable port 161 and 162 to use SNMP!**

Enable ports 161 and 162 in the WBM in menu “port”, so that the fieldbus coupler/controller can be reached via SNMP. The port numbers cannot be modified.

Note

**Modify parameter via WBM or SNMP objects!**

However, parameters that can be set on the html pages can also be changed directly by the appropriate SNMP objects.

Information

**Additional Information:**

Additional information for SNMP, the Management Information Base (MIB) and traps (event messages via SNMP) may be obtained from section “Fieldbus Communication” > ... > “SNMP (Simple Network Management Protocol).”

Note that the settings for SNMPV1/V2c and SNMPV3 are separate from each other: The different SNMP versions can be activated or used in parallel or individually on a fieldbus controller.

10.5.1 SNMP V1/V2c

The SNMP version 1/2c represents a community message exchange. The community name of the network community must thereby be specified.

Web-based Management



SNMP Configuration

This page is dedicated to the SNMP configuration. The new configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

SNMP Configuration

Description	WAGO 750-831 PFC ETHERNET
Physical location	LOCAL
Contact	support@wago.com

SNMP v1/v2c Manager Configuration

Protocol Enable	SNMP V1/V2c ☒
Local Community Name	public

SNMP v1/v2c Trap Receiver Configuration

Trap Receiver 1	0.0.0.0	
Community Name 1	public	
Trap Version	V1 ☒	V2 ☐
Trap Receiver 2	0.0.0.0	
Community Name 2	public	
Trap Version	V1 ☒	V2 ☐

UNDO
SUBMIT

Figure 75: WBM page “SNMP”

Table 63: WBM page “SNMP”

SNMP Configuration		
Entry	Value (Default)	Description
Description	WAGO 750-831 BACNET/IP	Device description (sysDescription)
Physical location	LOCAL	Location of device (sysLocation)
Contact	support@wago.com	E-mail contact address (sysContact)
SNMP v1/v2 Manager Configuration		
Entry	Value (Default)	Description
Protocol Enable	SNMP V1/V2c ☒	☒ Activating SNMP Version 1/2c
		☐ Deactivating SNMP-Version 1/2c
Local Community Name	public	Used community name
SNMP v1/v2 Trap Receiver Configuration		
Entry	Value (Default)	Description
Trap Receiver 1	0.0.0.0	IP address of 1. used SNMP manager
Community Name 1	public	1. Community name of the network community used
Trap Version	V1 ☒	V1 ☒ V2 ☐ Activating Traps Version 1
	V2 ☐	V1 ☐ V2 ☒ Activating Traps Version 2
Trap Receiver 2	0.0.0.0	IP address of 2. used SNMP manager
Community Name 2	public	2. Community name of the network community used
Trap Version	V1 ☒	V1 ☒ V2 ☐ Activating Traps Version 1
	V2 ☐	V1 ☐ V2 ☒ Activating Traps Version 2

10.5.2 SNMP V3

In SNMP version 3, exchanging messages is user-related. Each device, that knows the passwords set via WBM, may read or write values from the fieldbus coupler/-controller.

In SNMP V3, user data from SNMP messages can also be transmitted in encoded form. This is why SNMP V3 is often used in safety-related networks.

Web-based Management

SNMP Configuration

This page is dedicated to the SNMP configuration. The new configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.
'Authentication Key' and 'Privacy Key' have to be at least 8 characters.

SNMP v3 (user based)

1.User	activate ☐
Authentication Type	None ☐ MD5 ☒ SHA1 ☐
Security Authentication Name	SecurityName
Authentication Key	AuthenticationKey
Privacy Enable	DES ☒
Privacy Key	PrivacyKey
Notification/Trap enable	V3 ☐
Notification Receiver IP	0.0.0.0
2.User	activate ☐
Authentication Type	None ☐ MD5 ☒ SHA1 ☐
Security Authentication Name	SecurityName
Authentication Key	AuthenticationKey
Privacy Enable	DES ☒
Privacy Key	PrivacyKey
Notification/Trap enable	V3 ☐
Notification Receiver IP	0.0.0.0

Figure 76: WBM page “SNMP V3”

Table 64: WBM Page “SNMP V3”

SNMP v3 (user based)		
Entry	Value (Example)	Description
1. User / 2. User	activate ☒	☒ Activating user 1 or 2
		☐ Deactivating user 1 or 2
Authentication Type	None ☐ MD5 ☒ SHA1 ☐	None ☒ MD5 ☐ SHA1 ☐ No encryption of the authentication
		None ☐ MD5 ☒ SHA1 ☐ Encryption of the authentication with MD5
		None ☐ MD5 ☐ SHA1 ☒ Encryption of the authentication with SHA1
Security Authentication Name	Security Name	Enter the name, if the “authentication type” MD5 or SHA1 has been selected
Authentication Key	Authentication Key	Enter the password with at least 8 characters, if “authentication type” MD5 or SHA1 has been selected
Privacy Enable	DES ☒	☒ Activate the DES encryption of the data
		☐ Deactivate the DES encryption of the data
Privacy Key	Privacy Key	Enter the password of at least 8 characters in the encryption with DES
Notification/Trap enable	V3 ☒	☒ Activate the notification traps of the SNMP version 3
		☐ Deactivate the notification traps of the SNMP version 3
Notification Receiver IP	192.168.1.10	IP address of the notification manager

Two independent SNMPv3 users can be defined and activated via the html page (user 1 and user 2).

10.6 Watchdog

Click the link "Watchdog" to go to a Web site where you can specify the settings for the connection and MODBUS watchdog.

Web-based Management



Watchdogs

This page is for the configuration of the watchdogs. The configuration is stored in an EEPROM. Changes of the Connection Time will take effect immediately. Changes of the Modbus Watchdog will take effect after the next software or hardware reset. For more information see the manual.

Connection Watchdog

Connection Timeout Value (100ms):

Modbus Watchdog

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

Figure 77: WBM Page "Watchdog"

Table 65: WBM Page "Watchdog"

Connection watchdog		
Entry	Default	Description
Connection Timeout Value (100 ms)	600	Monitoring period for TCP links. After the completion of this period without any subsequent data traffic, the TCP connection is closed.
Modbus Watchdog		
Entry	Default	Description
State Modbus Watchdog	Disabled	Enabled – Watchdog is activated Disabled – Watchdog is disabled
Watchdog Type	Standard ☒	The set coding mask (watchdog trigger mask) is evaluated to determine whether the watchdog time is reset.
	Alternative ☐	The watchdog time is reset by any Modbus/TCP telegram.
Watchdog Timeout Value (100 ms)	100	Monitoring period for Modbus links. After the completion of this period without receiving a Modbus telegram, the physical outputs are set to "0".
Watchdog Trigger Mask (F 1 to F16)	0xFFFF	Coding mask for certain Modbus telegrams (Function Code FC1 ... FC16)
Watchdog Trigger Mask (F17 to F32)	0xFFFF	Coding mask for certain Modbus telegrams (Function Code FC17 ... FC32)

10.7 Clock

Specify the settings for the internal real-time clock on the “Clock” HTML page. Here, enter the current time and date and also select standard or daylight saving time.

Note



Reset the internal clock after a specified time without power supply!

The internal clock must be reset on initial startup or after the end of days, specified in the technical Data without power supply (Powerfail RTC Buffer). If the clock is not set, the clock begins with the date 01.01.2000 around 0:00 clock with time measurement.

Note



Daylight Saving Time (DST) change via function block or automatically!

To switch between standard and daylight-saving time via the web-based management system, select on page “Clock” the “Automatic (USA / EU)” option or optional “Manual”. To ensure that all functions are correctly carried out properly and at the right time even if the “Manual” option is selected, you can also obtain automatic conversion means of the function block “PrgDaylightSaving”. Exclusively in case of the “Manual” option it is allowed to use this function block, which is available in WAGO-I/O-PRO by integration of the library “DaylightSaving.lib”.

Note



Error message in WAGO I/O CHECK is possible after a power failure!

If you are using the software “WAGO-I/O-CHECK” after a loss of power has occurred, error messages may be generated. Should this occur, call up the Web-based management system and set the actual time under “Clock”. Then, call up the “WAGO-I/O-CHECK” program again

Note



Loss of telegrams possible when performing configuration during ongoing operation!

Telegrams may be lost if configuration is performed using WAGO-I/O-CHECK while the system is in operation.

Note



Use a WAGO RTC module for time synchronization!

You can also use a WAGO 750-640 RTC Module for your node to utilize the actual encoded time (Real-time – RTC) in your higher-level control system. An even greater degree of accuracy is achieved via RTC module than that obtained using the real-time clock in the coupler or controller.

Web-based Management



Clock configuration

Configuration Data

Time on device	11:07:54
Date (YYYY-MM-DD)	2015-11-19
Timezone (+/- hour:minute)	+1:00 ▾
Daylight Saving Time (DST)	☒ Automatic (USA) Second sunday in March 02:00 AM first sunday in November 02:00 AM
	☐ Automatic (EU) Last sunday in March 01:00 UTC last sunday in October 01:00 UTC
	☐ Manual: ☐
12 hour clock	☐

Figure 78: WBM Page “Clock”

Table 66: WBM Page "Clock"

Configuration Data			
Entry	Default	Value (example)	Description
Time on device	Coordinated Universal Time UTC	13:00:31	Set current time
Date (YYYY-MM-DD)	Date based on UTC	2013-02-08	Set current date
Time zone (+/- hour)	0	1 (MEZ)	Set time zone offset from the Coordinated Universal Time (UTC)
Daylight Saving Time (DST)	☒ Automatic (USA)	☒	☒ Enable automatic Daylight Saving Time (USA) (second Sunday in March, clocks are advanced from 02:00 AM to 03:00 AM / first Sunday in November, clocks are set back from 02:00 AM to 01:00 AM)
			☐ Disable automatic Daylight Saving Time (USA)
	☐ Automatic (EU)	☐	☒ Enable automatic Daylight Saving Time (EU) (last Sunday in March, clocks are advanced from 02:00 AM to 03:00 AM / last Sunday in October, clocks are set back from 03:00 AM to 02:00 AM)
			☐ Disable automatic Daylight Saving Time (EU)
	☐ Manual:	☒	☒ Manually enable summer time ☐ Manually enable winter time
		☐	Disable manual summer/winter time
12 hour clock	☐	☒	☒ Enable 12-hour display
			☐ Enable 24-hour display

10.8 Security

Use the “Security” HTML page with passwords to set up read and/or write access for various user groups to protect against configuration changes.

Note



Passwords can only be changed by “admin” and after software reset!

The “admin” user and associated password are required to change passwords.

Press the [**Software Reset**] button to restart the software for the setting changes to take effect.

Note



Note password restrictions!

The following restriction is applied for passwords:

- Max. 32 characters inclusive special characters.
-

Web-based Management

Security

This page is intended to disable the basic authentication. Additionally you can set new passwords for the existing user. The new values are stored in an EEPROM and changes will take effect immediately.

Webserver Security

Webserver authentication enabled ☒

UNDO SUBMIT

Webserver, FTP and PLC User configuration

User: guest Password:
Confirm Password:

UNDO SUBMIT

PLC Security

Port 2455 authentication enabled ☐

UNDO SUBMIT

Attention: You will lose the connection to the webserver after the software reset, if the IP configuration was changed. Please load the webpage with the proper address in this case again.

Software Reset

Figure 79: WBM Page "Security"

Table 67: WBM Page “Security”

Webserver Security		
Entry	Default	Description
Webserver authentication enabled	☒	☒ Enable password protection to access the Web interface
		☐ Disable password protection to access the Web interface
Webserver, FTP and PLC User configuration *)		
Entry	Default	Description
User	*)	Select admin, guest or user
Password	*)	Enter password
Confirm password		Enter password again to confirm
PLC Security		
Entry	Default	Description
Port 2455 authentication enabled	☒	☒ Enable password protection to access to the PLC program.
		☐ Disable password protection to access to the PLC program.

*) The following default groups exist:

User: admin	Password: wago
User: guest	Password: guest
User: user	Password: user

Note



Renew access after software reset!

If you initiate a software reset on this page, then the fieldbus coupler/controller starts with the configurations previously loaded into the EEPROM and the connection to the browser is interrupted.

If you changed the IP address previously, you have to use the changed IP address to access the device from the browser.

You have not changed the IP address and performed other settings; you can restore the connection by refreshing the browser.

10.9 MODBUS

Use the “MODBUS” HTML page to specify the settings for the MODBUS protocol.

Web-based Management



MODBUS Configuration

This page is for the configuration of Modbus features. The configuration is stored in an EEPROM. Changes of the multicast setup will take effect after the next software or hardware reset. Changes of Modbus Configuration Registers and Modbus flag register blocking will take effect immediately after submit.

Modbus UDP Multicast Address Setup

Enable Multicast ☐

MCAST Address 1:

MCAST Address 2:

MCAST Address 3:

MCAST Address 4:

MCAST Address 5:

Valid address range:
225.0.0.0 to 238.255.255.255

Modbus Configuration Registers

Range	Enabled
0x1028 - 0x1037:	☒
0x2040 - 0x2043:	☒

Blocked Modbus Flag Registers

Range: 0x3000 - 0x5FFF

	Range Start	Range End	Enabled
1	0x0000	0x0000	☐
2	0x0000	0x0000	☐
3	0x0000	0x0000	☐
4	0x0000	0x0000	☐
5	0x0000	0x0000	☐

Figure 80: WBM Page “Modbus”

Table 68: WBM Page “Modbus”

Modbus UDP Multicast Address Setup		
Entry	Default value	Description
Enable Multicast	☐	☒ Enable Multicast for MODBUS UDP transmission. In addition to its own IP address, the fieldbus controller receives MODBUS commands for the following registered MCAST addresses.
		☐ Multicast for MODBUS UDP transmission is not enabled. The fieldbus controller receives MODBUS commands only for the own IP address.
MCAST Address 1 ... 5:	0.0.0.0	Multicast address 1... 5, for the multicast will be enabled. The valid address range is shown in the WBM. Multiple assigned addresses are not valid.
Modbus Configuration Registers		
Entry	Default value	Description
Range 0x1028 – 0x1037	☒	☒ Enable MODBUS configuration register range.
Range 0x2040 – 0x2043	☒	☐ Disable MODBUS configuration register range.
Blocked Modbus Flag Registers		
Entry 1 ... 5	Default value	Description
Range Start	0x0000	Start address for the blocked ModbusTCP register range.
Range End	0x0000	End address for the blocked ModbusTCP register range.
Enabled	☐	☒ Requests to the blocked addresses of ModbusTCP flag register range are answered with the exception code 0x02 "Illegal Data Address". If a request is partially in the valid and partially in the restricted range, this is also answered with an exception.
		☐ The respective table entry is not considered.

Note



Multicast function only with a valid MCAST address active!

Activate the function “**Enable Multicast**”, without you enter an address not equal to 0.0.0.0, the function is automatically deactivated, after a click on the **[SUBMIT]** button.

If the function “**Enable Multicast**” is already enabled with valid addresses, you can not describe this MCAST address fields with invalid addresses, because of a click on the **[SUBMIT]** button resets to the last valid addresses. Thereby, the function “**Enable Multicast**” maintains its State.

10.10 BACnet

Use the “BACnet” HTML page to set the transmission rate of the internal data bus and the UDP port.

Web-based Management

BACnet

This page is for the configuration of BACnet. The configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

Configuration

Disable BACnet fieldbus	☐
UDP Port	47808
Who-Is online interval time (seconds)	60
Save trendlog files on removable disk	☐
Disable PLC online change	☒

Figure 81: WBM Page “BACnet”

Table 69: WBM Page “BACnet”

Configuration			
Entry	Default value	Description	
Disable BACnet fieldbus	☐	☒	The BACnet fieldbus is disabled. BACnet communication is no longer possible. BACnet-specific blinking codes are no longer displayed.
		☐	BACnet fieldbus is enabled
UDP Port	47808	UDP port for the BACnet/IP protocol	
Who-Is online interval time (seconds)	60	Minimum time interval between two “Who-Is” queries of a fieldbus controller for online identification of the communication partner.	
Save trendlog files on removable disk	☒	☒	Save trendlog file to SD card
		☐	Save trendlog file to internal flash

Table 70: WBM Page “BACnet”

Entry	Default value	Description
Disable PLC online change	☒	☒ Deactivating the “Online Change” function. The “Online change” function requires a great deal of RAM and is not useful for execution of BACnet functions. This function is therefore deactivated as a default in the 750-831.
		☐ Activating the “Online Change” function. This function reduces the maximum number of available BACnet objects. It can only be used together with a dedicated target file, which is not included in the standard delivery.  Note The “Online Change” function has no effect on BACnet object changes! Please note that the “Online Change” function for changing BACnet objects is not supported. The “Online Change” function cannot be used to change or delete BACnet objects or properties in the application on your fieldbus controller. To update the application, stop the runtime system, load the application again and perform a reset.



Note

Network number entry not necessary for WAGO BACnet Controllers!

A BACnet network number is used to enable a BACnet router to forward messages from one BACnet network to another. The network number is not used for communication between devices in the same network. The network numbers of devices from other networks are automatically identified by the WAGO BACnet Controller and used for communication. For this reason, the network number cannot be set on WAGO BACnet controllers. This only has to be set on the BACnet router used.

10.11 PLC Info

The WBM page “PLC Info” contains information about the current CODESYS project. First, this information has to be registered in the CODESYS project under the menu “Project” → “Project Information”.

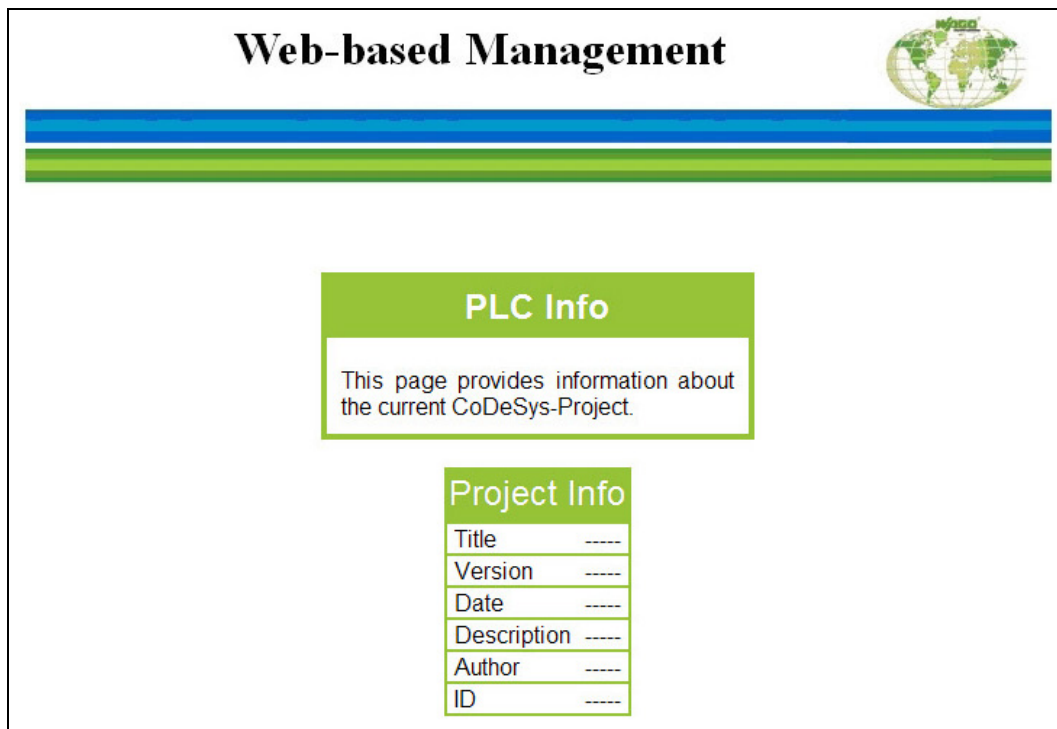


Figure 82: WBM Page "PLC Info"

Table 71: WBM Page “PLC Info”

Project Info			
Entry	Default value	Value (Example)	Description
Title	_____	SSL Client Example	Name of project
Version	_____	1.0.0	Project version
Date	_____	15.05.2012 08:50:27	Date and Time of the project
Description	_____	Test client to establish SSL connections	Description
Author	_____	JW	Project author
ID	_____	70632	Project ID
Checksum	No checksum available		Checksum of the boot project on the fieldbus controller. The checksum is formed from the read-out data as unsigned long 32-bit number in little-endian format and displayed in decimal. If the file "default.chk" is not available, "No checksum available" appears.

10.12 PLC

Click the “PLC” link to access a Web site where you can define the PLC functionality settings for your controller.

Web-based Management

PLC Configuration

This page is for the configuration of the PLC. The configuration is stored in an EEPROM. Changes will take effect after the next software or hardware reset. For more information see the manual.

PLC Features		
Function	Description	Enabled
Process image	Set outputs to zero, if user program is stopped.	☒
WebVisu	Set 'webvisu.htm' as default.	☐
	Open 'webvisu.htm' in frame.	☐
	Open 'webvisu.htm' in new window.	☒
I/O configuration	Compatible handling for ea-config.xml	☐
	Insert monitoring entries into ea-config.xml	☐

UNDO
SUBMIT

Attention! Please keep in mind that after changing the PLC root location, the boot project and all the retain/persistent memory will be lost after power on.

PLC Root Location

Internal File System

External SD memory card

UNDO
SUBMIT

Figure 83: WBM Page “PLC Settings”



Note

Return to “WebVisu.htm” page via the IP address of the fieldbus controller!

The “Webvisu.htm” page does not have any hyperlinks to the other Web sites. To deactivate this starting page function, or to go to other pages using hyperlinks, enter the IP address for your controller and the address for the original starting page in the URL line of your browser with the following syntax:
<http://IP address of your controller/webserv/Index.ssi>

Table 72: WBM Page “PLC”

PLC Features		
Function	Default	Description
Process image	Set outputs to zero, if user program is stopped ☒	☒ Activate, if all outputs must be set at zero when stopping the user program
		☐ Disable, if all outputs must remain at the last current value when stopping the user program
WebVisu	Set 'webvisu.htm' as default ☐	☒ Activate, if the page “Webvisu.htm” must be opened as starting page when calling up WMB instead of the standard starting page “Status Information”
		☐ Activate, if the standard starting page “Status Information” must be opened when calling up WMB
	Open 'webvisu.htm' in frame ☐	☒ Activate, if the page “Webvisu.htm” must be opened in the same frame
		☐ Activate, if the page “Webvisu.htm” must be opened in another frame
	Open 'webvisu.htm' in new window ☒	☒ Activate, if the page “Webvisu.htm” must be opened in the same window
		☐ Activate, if the page “Webvisu.htm” must be opened in another window

Table: WBM Page "PLC"

PLC Features			
Function	Default	Description	
I/O configuration	Compatible handling for ea-config.xml ☐	Activate, if the write authorizations must be assigned to the outputs of all bus terminals based on an existing file "ea-config.xml". Here, note whether a control system configuration has already been created and, if so, whether this configuration is correct or incorrect (see the following table). The current process values are displayed on the website "IO config", in addition to the displayed data channels. ☒ Correct control system configuration has been created in the project	
		Disable, if the write authorizations must be assigned to the outputs of all bus terminals of the PLC Here, note whether a control system configuration has already been created and, if so, whether this configuration is correct or incorrect (see the following table). ☐ Incorrect control system configuration has been created in the project	
		I/O configuration (function activated)	I/O configuration (function deactivated, standard setting):
		Writing privileges to the outputs of all modules are assigned on the basis of an existing ea-config.xml. The ea-config.xml file must be completely error-free; otherwise the writing privileges for all modules will be assigned to the standard fieldbus.	The outputs for all modules are assigned to the PLC. Any ea-config.xml file that may already be present is ignored and overwritten.
		Correct control system configuration has been created in the project	Writing privileges to the module outputs is taken from the control system configuration. A corresponding ea-config.xml file is generated in the file system.
		Incorrect control system configuration has been created in the project	The standard fieldbus is granted writing privileges to the outputs of all the modules.
	Insert monitoring entries into ea-config.xml ☐	Activate to also display the current process values on the html page "IO config" for the displayed data channels. ☒ Correct control system configuration has been created in the project	
		Disable, if no process values must be displayed on the html page "IO config". ☐ Incorrect control system configuration has been created in the project	

Table 73: WBM Page „PLC“

PLC Root Location		
Function	Default value	Description
Internal File System	☒	Activate this function if all of the files related to the boot project and the WAGO-I/O-PRO application are to be saved to the internal file system in the future (A:\PLC). ☒ No files are copied from the SD card to the internal file system. The BACnet configuration stored on the internal drive will be used.
		☐ This is deactivated when the option "External SD memory card" is selected.
External SD memory card	☐	Activate this function if all of the files related to WebVisu, the boot project and the WAGO-I/O-PRO application are to be saved to the SD card in the future (S:\PLC). ☒ No files are copied from the internal file system to the SD card. The BACnet configuration stored on the SD card will be used.
		☐ This is deactivated when the option "Internal File System" is selected.

Note



Provide sufficient storage space for PLC folders!

The following can occur when there is less free storage space on the SD card than is normally required for the PLC folder and the files it contains:

- The files "webvisu.htm" and "error_ini.xml" are created without any content, meaning that the WBM page "Web Visu" is blank.
- The files "webvisu.htm" and "error_ini.xml" are created but are incomplete.

In either case, this condition remains until the entire "PLC" folder is deleted and sufficient storage space made available.

The function "PLC Root Location" remains set to "External SD memory card" and the MS/BT LED flashes red.

Note



Only the "PLC" folder is affected by the switchover!

Note that the selection of the data storage location is always limited to the "PLC" directory, while all other directories and the WBM pages retain their settings. The "PLC" directory is not deleted.

A distinction must be drawn between the following cases when changing the options for storage from the internal file system to storage on the SD card:

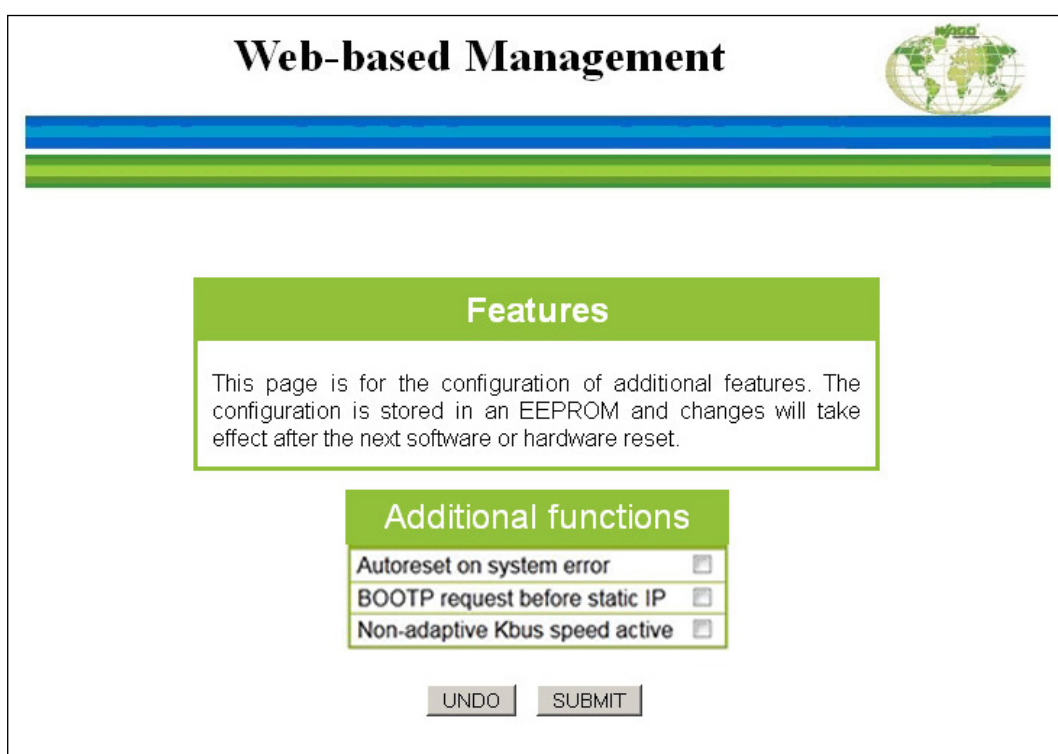
- **SD card not inserted:**
You have selected the option "**External SD memory card**" and clicked the button **[SUBMIT]**. Error code 10-1 is output via the IO LED.
- **Empty SD card inserted:**
You have selected the option "**External SD memory card**", clicked the button **[SUBMIT]** and have restarted the fieldbus controller; the directory "**S:\PLC**" is created automatically on the SD card and the standard files "**webvisu.htm**" and "**error_ini.xml**" are extracted to there.
- **Inserted SD card is already written:**
You have selected the option "**External SD memory card**" and clicked the button **[SUBMIT]**. The PLC program is started from the SD card, provided a boot project exists on the SD card and the start/stop switch is at the proper position.

10.13 Features

Use the “Features” HTML page to enable or disable additional functions.

The function "Autoreset on system error" can ensure safe, reliable and continuous operation when activated for areas that are difficult to access (e.g. closed rooms, equipment centers on building roofs). An autorestart is executed as soon as the controller has an error status that requires a restart.

With the original factory settings, this function is deactivated (default), meaning that diagnostics is indicated via the blink code for the I/O LED when an error occurs. A manual restart must then be conducted after error evaluation and rectification.



Web-based Management

Features

This page is for the configuration of additional features. The configuration is stored in an EEPROM and changes will take effect after the next software or hardware reset.

Additional functions	
Autoreset on system error	☐
BOOTP request before static IP	☐
Non-adaptive Kbus speed active	☐

UNDO SUBMIT

Figure 84: WBM Page “Features”

Table 74: WBM Page “Features”

Additional functions		
Entry	Default	Description
Autoreset on system error	☐	☒ enables an automatic software reset to be conducted when a system error occurs
		☐ disables an automatic software reset to be conducted when a system error occurs
BootP Request before Static-IP	☐	Automatically set the static IP address enabled. ☒ For this configuration, the fieldbus coupler/controller uses a statically configured IP address if the request via BootP fails.
		Automatically set the static IP address disabled. ☐ For this configuration, the IP address request via BootP is repeated in the event of error.
Non-adaptive Kbus speed active	☐	Disable the rate adjustment of the internal bus for short fieldbus nodes. With short fieldbus nodes, doing so makes more computing power available for the PLC application. The internal bus rate is correspondingly lower. ☒
		☐ Enables the rate adjustment of the internal bus.

10.14 I/O Config

Click the link “I/O Config” to view the configuration and/or write access privileges for the outputs of your fieldbus node.

The node structure created using the “WAGO-I/O-PRO I/O Configurator” hardware configuration tool is displayed in the window. If no modules are shown in this window, no hardware configuration and, thus, no allocation of write access privileges have been assigned. In this case, the handling defined at the Web site “PLC” by the function “I/O configuration - Compatible handling for ea-config.xml” will be applied to assign the write privileges for all outputs either to the standard fieldbus, or to the PLC.

Configuration details	
Number of modules on terminalbus	2
Number of modules in I/O configuration	0

create ea-config.xml

I/O configuration file			
Pos	Module	Type	Mapping

Figure 85: WBM Page “I/O config” (View 1)



Information

Additional Information

For more detailed information about the WAGO-I/O-PRO I/O Configurator, refer to the Section “Startup of Fieldbus Node”.

When the function “I/O configuration Insert monitoring entries into ea-config.xml” is also activated at the Web site “PLC”, the current process values will also be shown for the data channels that are displayed.

Web-based Management



I/O configuration

Configuration details

Number of modules on terminalbus	3
Number of modules in I/O configuration	0

create ea-config.xml

I/O configuration file

Pos	Module	Type	Mapping
1	750-4xx	2DI	Fieldbus 3
2	750-642/000-000	1COMPLEX	Fieldbus 3
3	750-476/000-000	2AI	Fieldbus 3

Figure 86: WBM Page “I/O config” (View With Process Values)

Table 75: WBM Page “I/O Config”

Configuration details		
Entry	Value (Example)	Description
Number of modules on terminal bus	5	Number of I/O modules (hardware)
Number of modules in I/O configuration	5	Number of I/O modules in the hardware configuration of the I/O
[create ea-config.xml]	-	Writes the current bus terminal structure and the field bus assignment in the „ea-config.xml” file.
I/O configuration file		
Entry	Value (Example)	Description
Pos	1	Position of the I/O module in the hardware
Module	750-4xx M001Ch1 M001Ch2	Product number of the integrated I/O module M = module, 001 = position 1, Ch1 = channel 1 M = module, 002 = position 2, Ch2 = channel 2
Type	2DI	I/O module type, e.g. 2 DI (2 Channel Digital Input Module)
Mapping	Fieldbus 3	Mapping via PLC, fieldbus 1 etc. (Entries depend on the coupler/controller, see WAGO-I/O-PRO under control parameters/module parameters)



Note

Enter I/O modules in the I/O Configurator!

Enter the I/O modules used in the I/O configurator of WAGO-I/O-PRO. Here, open the **Control Configuration** in the **Resources** register and add your I/O modules to the I/O module figure.

The added I/O modules must match the hardware in sequence and quantity. The entries “Number of modules on terminalbus” and “Number of modules in I/O configuration” on the html page “I/O Config” serve as control.

10.15 Disk Info

Information about the internal drive is given on the “Disk Info” page. Additional information about the SD card is also displayed there when the SD card is inserted.

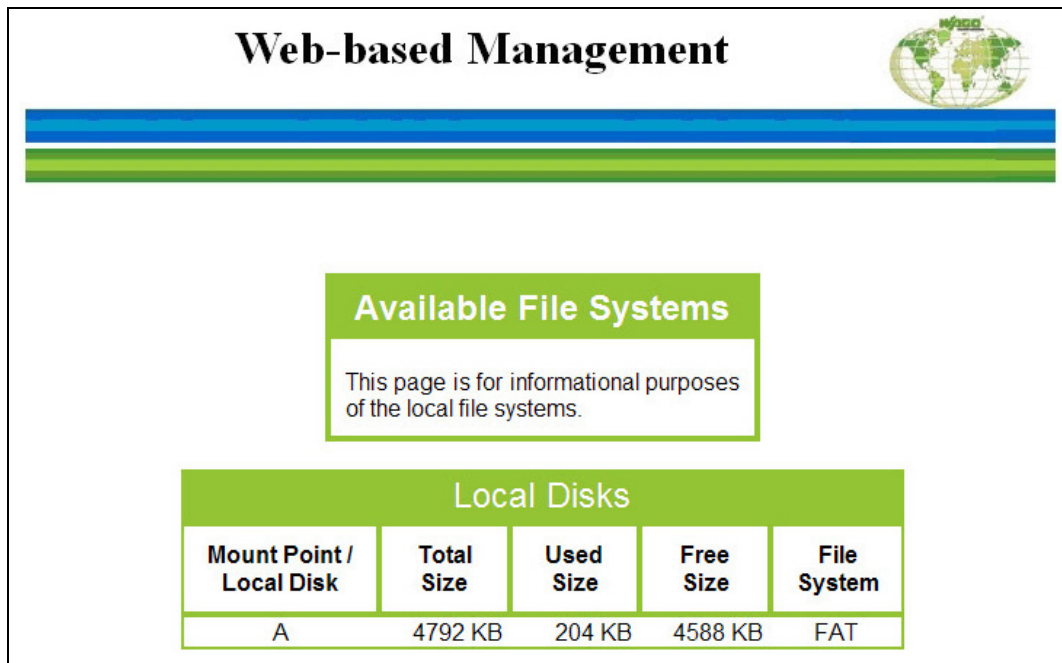


Figure 87: WBM Page “Disk Info”

Table 76: WBM Page “Disk Info”

Local Disks		
Entry	Value (Example)	Description
Mount Point/Local Disk	A	Directory - A: internal memory - S: removable disk
Total Size	1868 KB	Total size of the file system
Used Size	172 KB	Used memory capacity
Free Size	1696 KB	Free memory capacity
File System	FAT	File system (File Allocation Table)

10.16 SD Card

On the “SD Card” HTML page, you can find information about the memory card used, as well as delete the content of the “PLC” folder or create the “PLC” folder on the memory card via two buttons.

Web-based Management

SD Card

This page contains SD card info and tools.

SD Card Info

Description	Value
Card Serial Number:	76169405
Card Name:	SAMSUNG_8GB
Free card space:	7.4 GB
Configuration file S:\settings\750-8xx.xml is present.	

Tools

Description	Command	Status
Delete PLC folder contents	START	READY
Create PLC folder in SD root directory	START	READY

Figure 88: WBM Page “SD Card”

Table 77: WBM Page “SD Card”

SD Card Info			
Entry	Default	Value (example)	Description
Card Serial Number	_____	76169405	Serial number of the memory card
Card Name	_____	SAMSUNG_ 8 GB	Designation of the memory card
Free Card Space	_____	7.4 GB	Available space on the memory card
		Configuration file S:\settings\750-88x.xml is present.	Status message if the configuration file is on the memory card.
Tools			
Entry	Button	Status (Example)	Description
Delete PLC folder contents	[START]	READY	Delete the content in the “PLC” folder
Create PLC folder in SD root directory	[START]	READY	Create the “PLC” folder in the root directory of the memory card if it does not already exist with subsequent extraction of the webvisu.htm and error_ini.xml default files.

10.17 Backup & Restore

On the “Backup & Restore” HTML page, configure the settings to back up and restore device data.

Use an SD card to transfer device data from one fieldbus controller to one or more other fieldbus controller.

Web-based Management

Backup & Restore

This page is for the configuration of backup & restore functionality.

Services

Command	Description	Status
START	Backup ALL device settings to removable disk (77 kB needed).	OK
START	Restore device settings from removable disk using the option settings below.	OK

Restore Option Setting

Setting	Description
☐	Autorestore Global Enable This option will be disabled after the next reboot if the following option is set to next reboot only
☒	Restore settings from disk on next reboot only
☐	Always restore settings from disk on reboot
☐	IP address, dip switch IP address, network mask, domain name, host name, gateway, DNS servers, auto reset on system error, BOOTP request before static IP, non-adaptive Kbus speed
☐	Ports, security, time server, multicast settings, blocked Modbus flag registers, MAC filter, Ethernet hardware settings
☐	PLC settings, PLC folder and subfolders, boot project, ETC folder and web visualisation
☐	Non-volatile memory
☐	File system
☐	Watchdog, trace settings, SNMP, clock settings

Figure 89: WBM Page “Backup & Restore”

Table 78: WBM Page "Backup & Restore"

Services		
Entry	Button	Description
Backup ALL device settings to removable disk (160 kB needed).	[START]	Save all device's internal data and settings to SD card. Under the entry "Status" the status signal is displayed.
Restore device settings from removable disk using the option settings below.	[START]	Taking into account the choosen settings, which are listed below under "Restore Option Setting", the device's internal data and settings loading from the SD card to the device. Under the entry "Status" the status signal is displayed.
Entry	Status Signal	Description
Status	OK	Storage completed (Backup).
	DONE	Loading completed (Restore).
	ERROR(1)	The source location for synchronization doesn't exist.
	ERROR(2)	The EEPROM settings have been modified.
	ERROR(10)	Memory card is full.
	ERROR(11)	There is no target memory left for function execution.
	ERROR(12)	The length of an absolute path exceeds the buffer size.
	ERROR(20)	No access to the source file.
	ERROR(21)	No access to the target file.
	ERROR(30)	An error occurred during reading directory content.
	ERROR(31)	An error occurred creating a (sub)directory.
	ERROR(32)	The source or target location for synchronization is invalid.
	ERROR(40)	No access to the EEPROM.
	ERROR(41)	The parameter block table is invalid.
	ERROR(42)	The file length is different from the size of the NOVRAM.
	ERROR(43)	No access to the NOVRAM file.
Restore Option Setting		
Entry	Value (Default)	Description
Autorestore Global Enable This option will be disabled after the next reboot if the following option is set to next reboot only	☒	☒ Device data is loaded from the SD card when restarting.
		☐ No device data is loaded from the SD card when restarting. If the „Only load at next restart” setting is enabled, this option is automatically disabled.
- Restore settings from SD card on next reboot only	☒	Data is only loaded from the SD card after the device is restarted.
- Always restore settings from SD card on reboot	☐	Data is loaded from the SD card each time the device is restarted.

IP address, dip switch IP address, network mask, domain name, host name, gateway, DNS servers, auto reset on system error, BOOTP request before static IP, non-adaptive Kbus speed	☒	☒ The most important startup settings such as IP addresses, etc. are loaded.
		☐ The specified IP settings are not loaded.
Ports, security, time server, multicast settings, blocked Modbus flag registers, MAC filter, Ethernet hardware settings	☒	☒ Protocol and network settings are loaded from the SD card.
		☐ The specified protocol and network settings are not loaded.
PLC settings, PLC folder und subfolders, boot project, ETC folder and web visualization	☒	☒ PLC settings, boot project, ETC files and pages of the Web-based Management System are loaded.
		☐ The specified settings are not loaded.
Non-volatile memory	☒	☒ Non-volatile memory is loaded.
		☐ Non-volatile memory is not loaded.
File system	☒	☒ User files and folders stored in the file system are loaded from the SD card.
		☐ No user files and folders stored in the file system are loaded from the SD card.
Watchdog, trace settings, SNMP, clock settings	☒	☒ Watchdog, trace, SNMP and time settings are loaded from the SD card.
		☐ No Watchdog, trace, SNMP and time settings are loaded from the SD card.

Note



Note about using a memory card with an empty “Backup” directory!

Please note the following before inserting a memory card with an empty "Backup" directory in the fieldbus controller:

If the PLC root location is set to “External SD memory card”, then disable the “Autorestore Global Enable” function before inserting the memory card and before switching on the fieldbus controller in WBM on the “Backup & Restore” page.

Otherwise, the I/O LED will output the error god 14/7 (“Restore function failed!”).

10.18 WebVisu

The visualization of your programmed application is displayed on the html page “WebVisu”, provided you have created it with the visualization editor in WAGO-I/O-PRO and loaded it into the controller.

Perform the following settings in WAGO-I/O-PRO, so that an html page with your visualization is automatically created at the transmission of your project:

1. Double click to open the **Target System Settings** in the **Resource** register.
2. Open the **Visualization** register.
3. Select the **Web Visualization** option with a hatch mark.
4. Confirm with **OK**.

A link is then created to this html page “WebVisu” by the Web-based Management system. You can set the html page “WebVisu” as the starting page.

1. Call up the page “PLC” in the web-based Management-System.
2. a.) To set the HTML page “WebVisu” as the start page, use the function **WebVisu – Set 'webvisu.htm' as default**. When accessing the web-based management system, the “WebVisu” page is opened instead of the default WBM start page “Information”. However, the links to switch to the other WBM pages is then no longer available.

Note



Returning to the “WebVisu.htm” page is only possible via the IP address of the fieldbus controller!

The “Webvisu.htm” page does not have any hyperlinks to other Web sites. To deactivate the starting page function again, or to go to other pages using hyperlinks, enter the IP address for your controller and the address for the original starting page in the URL line of your browser with the following syntax:

<http://IP address of your controller/webserve/Index.ssi>.

- b.) To call up the the HTML page “WebVisu” in an eternal window (default setting), use the function **WebVisu – Open 'webvisu.htm' in new window**. Clicking on the “WebVisu” link opens a new window that displays the HTML page with visualization of your configured application. The links to switch to the other WBM pages are still available with this setting.
- c.) To call up the HTML page “WebVisu” on the WBM site directly, use the function **WebVisu – Open 'webvisu.htm' in frame**. Clicking on the “WebVisu” link opens the HTML page with visualization of your configured application in a frame in the WBM window directly. The

links to switch to the other WBM pages are still available with this setting.

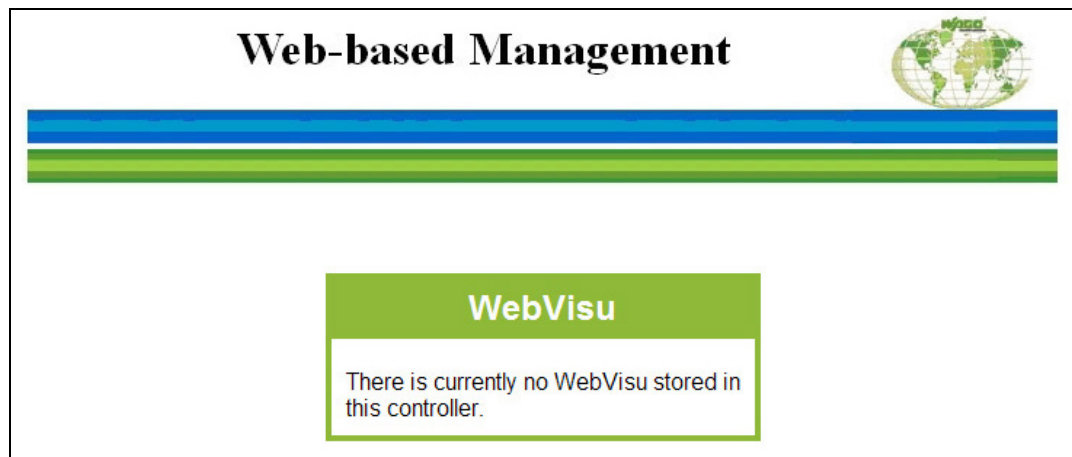


Figure 90: WBM Page “WebVisu”

11 Diagnostics

11.1 LED Signaling

For on-site diagnostics, the fieldbus controller has several LEDs that indicate the operational status of the fieldbus controller or the entire node (see following figure).

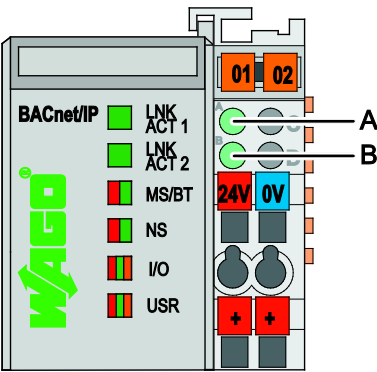


Figure 91: Display Elements

The diagnostics displays and their significance are explained in detail in the following section.

The LEDs are assigned in groups to the various diagnostics areas:

Table 79: LED Assignment for Diagnostics

Diagnostics area	LEDs
Fieldbus status	• LNK ACT 1 • LNK ACT 2 • MS/BT • NS
Node status	• I/O • USR
Status Supply Voltage	• A (system supply) • B (field supply)

11.1.1 Evaluating Fieldbus Status

The health of the ETHERNET Fieldbus is signaled through the top LED group ('LNK ACT 1, 2', 'MS/BT', und 'NS').

The two-colored LEDs 'MS/BT' (module status/BACnet) and 'NS' (network status) are used to display the status of the system and the fieldbus connections.

Table 80: Fieldbus Diagnostics – Solution in Event of Error

LED Status	Meaning	Solution
LNK ACT 1, 2		
green	The fieldbus node is connected to the physical network.	-
green flashing	The fieldbus node sends and receives Ethernet telegrams	-
off	The fieldbus node is not connected to the physical network.	1. Check the fieldbus cable.
MS/BT		
green	Network initialization completed, BACnet stack started. The fieldbus controller is ready for BACnet data traffic.	-
green flashing	Initialization of network and BACnet functions in progress.	1. Wait until initialization has been completed.
red	The system indicates a not remediable error	1. Restart the device by turning the power supply off and on again. 2. If the error still exists, please contact the I/O support.
red flashing	Error during initialization of the BACnet stack	1. Correct the BACnet configuration
off	No system supply voltage	1. Check the supply voltage.
NS		
green	At least one MODBUS/TCP connection is developed.	-
grün flashing	No MODBUS/TCP connection.	-
red	The system indicates a double IP-address in the network	1. Use an IP address that is not used yet.
red flashing	At least one MODBUS/TCP connection announced a Timeout, where the controller functions as target.	1. Restart the device by turning the power supply off and on again. 2. Develop a new connection.
red/green flashing	Self test	-
off	No IP address is assigned to the system.	1. Assign to the system an IP address by BootP, DHCP or the Ethernet Settings tool.

11.1.2 Evaluating Node Status – I/O LED (Blink Code Table)

The communication status between fieldbus coupler/controller and the I/O modules is indicated by the I/O LED.

Table 81: Node Status Diagnostics – Solution in Event of Error

LED Status	Meaning	Solution
I/O		
green	The fieldbus node is operating correctly.	Normal operation.
orange flashing	Start of the firmware. 1 ... 2 seconds of rapid flashing indicate start-up.	-
red	Fieldbus coupler/controller hardware defect	Replace the fieldbus coupler/controller.
red flashing	Flashing with approx.. 10 Hz indicates the initialization of the internal bus or of a internal bus error.	Note the following flashing sequence.
red cyclical flashing	Up to three successive flashing sequences indicate internal data bus errors. There are short intervals between the sequences.	Evaluate the flashing sequences based on the following blink code table. The blinking indicates an error message comprised of an error code and error argument.
off	No data cycle on the internal bus.	The fieldbus coupler/controller supply is off.

Device boot-up occurs after turning on the power supply. The I/O LED flashes orange.

Then the bus is initialized. This is indicated by flashing red at 10 Hz for 1 ... 2 seconds.

After a trouble-free initialization, the I/O LED is green.

In the event of an error, the I/O LED continues to blink red. Blink codes indicate detailed error messages. An error is indicated cyclically by up to 3 flashing sequences.

After elimination of the error, restart the node by turning the power supply of the device off and on again.

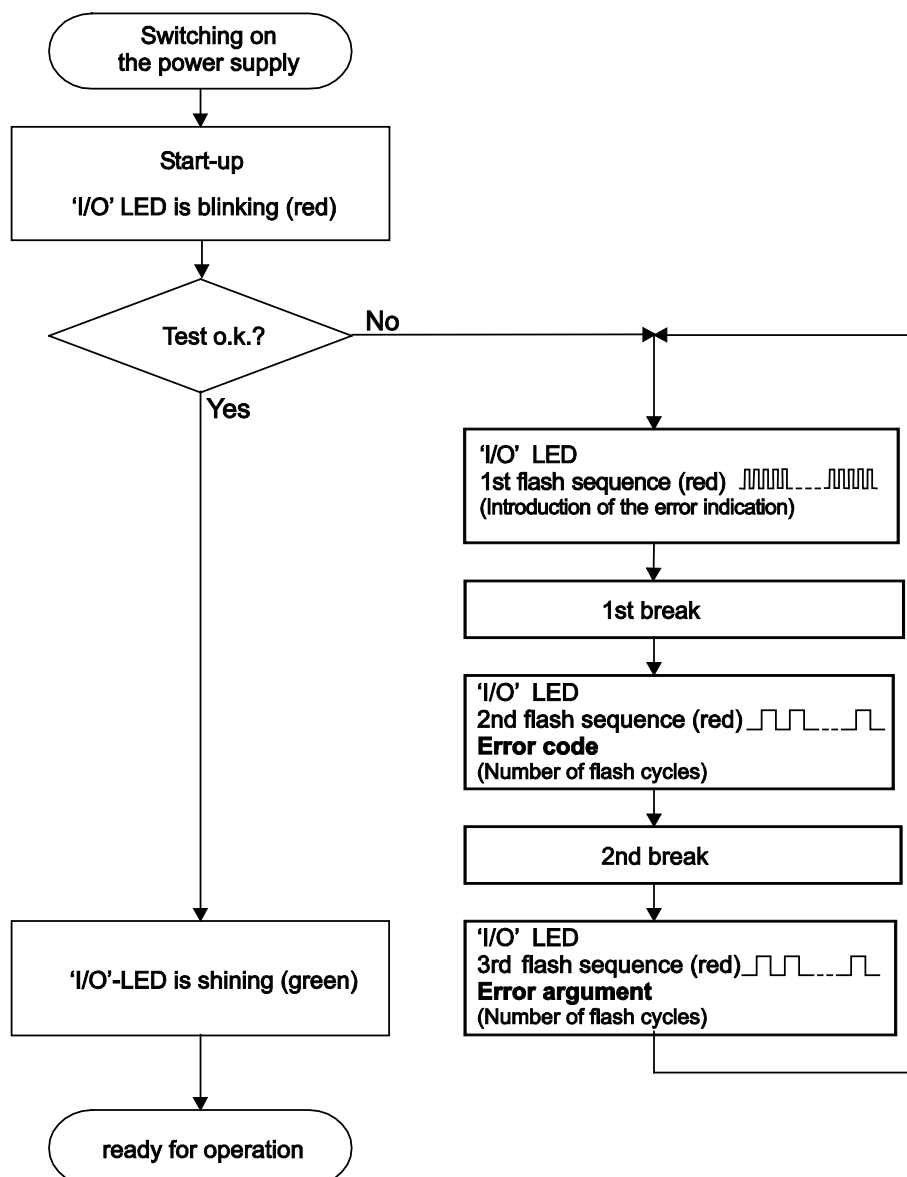


Figure 92: Node Status – I/O LED Signaling

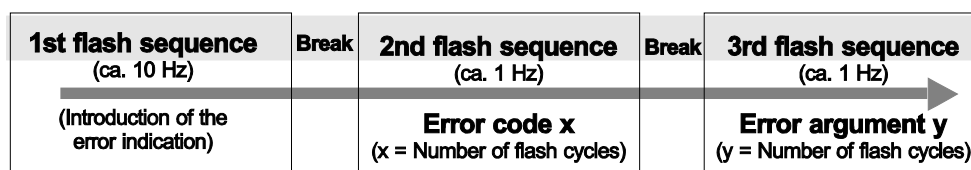


Figure 93: Error Message Coding

Example of a module error:

- The I/O LED starts the error display with the first flashing sequence (approx. 10 Hz).
- After the first break, the second flashing sequence starts (approx. 1 Hz): The I/O LED blinks four times. Error code 4 indicates “data error internal data bus”.

- After the second break, the third flashing sequence starts (approx. 1 Hz):
The I/O LED blinks twelve times.
Error argument 12 means that the internal data bus is interrupted behind the twelfth I/O module.

The thirteenth I/O module is either defective or has been pulled out of the assembly.

Table 82: Blink Code Table for the I/O LED Signaling, Error Code 1

Error code 1: "Hardware and configuration error"		
Error Argument	Error Description	Solution
1	Overflow of the internal buffer memory for the attached I/O modules.	1. Turn off the power for the node. 2. Reduce the number of I/O modules. 3. Turn the power supply on again. 4. If the error persists, replace the fieldbus controller.
2	I/O module(s) with unknown data type	1. Determine the faulty I/O module by first turning off the power supply. 2. Plug the end module into the middle of the node. 3. Turn the power supply on again. 4. - LED continues to flash? - Turn off the power supply and plug the end module into the middle of the first half of the node (toward the fieldbus controller). - LED not flashing? - Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller). 5. Turn the power supply on again. 6. Repeat the procedure described in step 4 while halving the step size until the faulty I/O module is detected. 7. Replace the faulty I/O module. 8. Inquire about a firmware update for the fieldbus controller.
3	Invalid check sum in the parameter area of the fieldbus controller.	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
4	Fault when writing in the serial EEPROM.	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
5	Fault when reading the serial EEPROM	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
6	The I/O module configuration after AUTORESET differs from the configuration determined the last time the fieldbus controller was powered up.	1. Restart the fieldbus controller by turning the power supply off and on.
7	Invalid hardware-firmware combination.	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.

Table 82: Blink Code Table for the I/O LED Signaling, Error Code 1

Error code 1: "Hardware and configuration error"		
Error Argument	Error Description	Solution
8	Timeout during serial EEPROM access.	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
9	Bus controller initialization error	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
10	Buffer power failure real-time clock (RTC)	1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
11	Fault during read access to the real-time clock (RTC)	1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
12	Fault during write access to the real-time clock (RTC)	1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
13	Clock interrupt fault	1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
14	Maximum number of gateway or mailbox modules exceeded	1. Turn off the power for the node. 2. Reduce the number of corresponding modules to a valid number.

Table 83: Blink Code Table for the I/O LED Signaling, Error Code 2

Error code 2: "Exceeded Process Image"		
Error Argument	Error Description	Solution
1	Not used	-
2	Process image is too large.	1. Turn off the power supply of the node. 2. Reduce number of I/O modules. 3. Turn the power supply on.

Table 84: Blink Code Table for the I/O LED Signaling, Error Code 3

Error code 3: "Protocol error, internal bus"		
Error Argument	Error Description	Solution
-	Internal data bus communication is faulty, defective module cannot be identified.	- Are passive power supply modules (750-613) located in the node? - 1. Check that these modules are supplied correctly with power. 2. Determine this by the state of the associated status LEDs. - Are all modules connected correctly or are there any 750-613 Modules in the node? - 1. Determine the faulty I/O module by turning off the power supply. 2. Plug the end module into the middle of the node. 3. Turn the power supply on again. 4. - LED continues to flash? - - Turn off the power supply and plug the end module into the middle of the first half of the node (toward the fieldbus controller). - LED not flashing? - - Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller). 5. Turn the power supply on again. 6. Repeat the procedure described in step 4 while halving the step size until the faulty I/O module is detected. 7. Replace the faulty I/O module. 8. If there is only one I/O module on the fieldbus controller and the LED is flashing, either the I/O module or fieldbus controller is defective. 9. Replace the defective component.

Table 85: Blink Code Table for the I/O LED Signaling, Error Code 4

Error code 4: "Physical error, internal bus"		
Error Argument	Error Description	Solution
-	Internal bus data transmission error or interruption of the internal data bus at the fieldbus controller	1. Turn off the power supply to the node. 2. Plug in an end module behind the fieldbus controller. 3. Turn the power supply on. 4. Observe the error argument signaled. - Is no error argument indicated by the I/O LED? - 5. Replace the fieldbus controller. - Is an error argument indicated by the I/O LED? - 5. Identify the faulty I/O module by turning off the power supply. 6. Plug the end module into the middle of the node. 7. Turn the power supply on again. 8. - LED continues to flash? - Turn off the power and plug the end module into the middle of the first half of the node (toward the fieldbus controller). - LED not flashing? - Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller). 9. Turn the power supply on again. 10. Repeat the procedure described in step 6 while halving the step size until the faulty I/O module is detected. 11. Replace the faulty I/O module. 12. If there is only one I/O module on the fieldbus controller and the LED is flashing, either the I/O module or fieldbus controller is defective. 13. Replace the defective component.
n*	Interruption of the internal data bus behind the nth bus module with process data, the maximum supported number is reached, the following modules are no longer supported.	1. Turn off the power supply of the node. 2. Reduce number of I/O modules. 3. Turn the power supply on.

* The number of light pulses (n) indicates the position of the I/O module.
I/O modules without data are not counted (e.g., supply modules without diagnostics)

Table 86: Blink Code Table for the I/O LED Signaling, Error Code 5

Error code 5: "Initialization error, internal bus"		
Error Argument	Error Description	Solution
n*	Error in register communication during internal bus initialization	1. Turn off the power supply to the node. 2. Replace the (n+1) I/O module containing process data. 3. Turn the power supply on.

* The number of light pulses (n) indicates the position of the I/O module.
I/O modules without data are not counted (e.g., supply modules without diagnostics)

Table 87: Blink Code Table for the I/O LED Signaling, Error Code 6

Error code 6: "Fieldbus specific errors"		
Error Argument	Error description	Solution
1	Invalid MACID	1. Turn off the power supply of the node. 2. Exchange fieldbus controller. 3. Turn the power supply on again.
2	Ethernet Hardware initialization error	1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, exchange the fieldbus controller.
3	TCP/IP initialization error	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. If the error still exists, exchange the bus coupler.
4	Network configuration error (no IP Address)	1. Check the settings of BootP server.
5	Application protocol initialization error	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. If the error still exists, exchange the bus coupler.
6	Process image is too large	1. Turn off the power supply of the node. 2. Reduce number of I/O modules
7	Double IP address in network	1. Change configuration. Use another IP address, which is not yet present in network. 2. Restart the fieldbus coupler by turning the power supply off and on again.
8	Error when building the process image	1. Turn off the power supply of the node. 2. Reduce number of I/O modules 3. Restart the fieldbus coupler by turning the power supply off and on again. 4. If the error still exists, exchange the bus coupler.
9	Error with mapping between bus modules and fieldbus	1. Check EA-Config.xml file on the fieldbus controller
10	The project in WAGO-I/O-PRO differs from the SYM_XML file or the SYM_XML file is missing. The connection between WAGO-I/O-PRO and BACnet is interrupted.	1. Check both the project in WAGO-I/O-PRO and the SYM_XML file. Both must match.

Table 88: Blink Code Table for the I/O LED Signaling, Error Code 7

Error code 7: "Not supported I/O module"		
Error Argument	Error Description	Solution
n	First unsupported I/O module in place of n.	1. Turn off the power supply to the node. 2. Replace the nth I/O module containing process data or reduce the number of modules to the number of n-1. 3. Turn the power supply on.

Table 89: Blink Code Table for the I/O LED Signaling, Error Code 8 ... 9

Error code 8 ... 9 –not used–		
Error Argument	Error Description	Solution
-	not used	-

Table 90: Blink Code Table for the I/O LED Signaling, Error Code 10

Error code 10: "PLC program fault"		
Error Argument	Error Description	Solution
1	Error when implementing the PFC run time system	1. Plug the SD card into the fieldbus controller. 2. Restart the fieldbus controller by turning the power supply off and on again. 3. If the error still exists, please contact the I/O Support.
2	Error when generating the PFC inline code	1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, please contact the I/O Support.
3	An IEC task exceeded the maximum running time or the sampling interval of the IEC task could not be kept (Watchdog)	1. Check the task configuration concerning the adjusted sampling intervals and watchdog times.
4	PFC Web-Visualization initialization error	1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, please accomplish a reset (origin) in WAGO-I/O-PRO. 3. Compile the project again. 4. Transfer the project to the controller.
5	Error when synchronizing the PLC configuration with the internal data bus	1. Check the information of the connected modules in the PLC configuration of WAGO-I/O-PRO. 2. Compare this information with the modules that are actually connected. 3. Compile the project again. 4. Transfer the project to the controller.

Table 91: Blink Code Table for the I/O LED Signaling, Error Code 11

Error code 11: "Gateway-/Mailbox I/O module fault"		
Error Argument	Error Description	Solution
1	Maximum number of Gateway modules exceeded	1. Turn off the power supply of the node. 2. Reduce number of Gateway modules. 3. Turn the power supply on again.
2	Maximum size of Mailbox exceeded	1. Turn off the power supply of the node. 2. Reduce the Mailbox size. 3. Turn the power supply on again.
3	Maximum size of process image exceeded due to the put Gateway modules	1. Turn off the power supply of the node. 2. Reduce the data width of the Gateway modules. 3. Turn the power supply on again.

* The number of blink pulses (n) indicates the position of the I/O module.
I/O modules without data are not counted (e.g. supply module without diagnosis)

Table 92: Blink code table for I/O LED signaling, error code 12

Error code 12 "Fault during saving to memory area of RAM"		
Error Argument	Error description	Remedy
3	RAM is full	1. Reduce number of BACnet objects. 2. Restart the access function.

Table 93: Blink Code Table for the I/O LED Signaling, Error Code 14

Error code 14: "Error in external storage (memory card)"		
Error Argument	Error Description	Solution
1	Memory card cannot be used (message from driver)	1. Replace the memory card with a new one. 2. Should this error persist, contact I/O Support..
2	Function cancelled by the user (30 s)	1. Plug in the memory card. 2. a) During the system start: Restart the fieldbus controller by switching the power supply off and on. b) In ongoing operation: Restart the access function.
3	Internal memory is full	1. Check the internal file system and the /copy/ directory on the memory card. The size of the /copy/ directory may not exceed the total size of the internal file system! 2. If necessary, delete files (e.g. through FTP, formatting the internal drive or cleaning the memory card on the PC).
4	Memory card is full	1. Use the PC to create storage space on the memory card or replace the memory card with a new one. 2. Restart the access function.
5	No access to EEPROM	1. Plug in the memory card. a) During system start: Start the fieldbus controller by switching the power supply off and on. b) In ongoing operation: Restart the access function. 2. Should this error persist, contact I/O Support.
6	Backup function failed	1. Remove the memory card and remove the write protection. 2. Plug the memory card back in. 3. Restart the access function. 4. If the error persists, replace the memory card with a new one. 5. Restart the access function. 6. Should the error persist, contact I/O Support.
7	Restore function failed	1. Plug the memory card back in if the memory card has been removed. 2. Close any existing FTP connection, establish a new connection, and restart the access function. 3. Should the error persist, restart the device and the access function. 4. If the error should persist, contact I/O Support.

11.1.2.1 USR LED

The bottom indicator LED (“USR”) is provided for visual output of information.

Control of the LED from the application program is conducted using the functions from the WAGO-I/O-PRO library “Visual.lib”.

11.1.3 Evaluating Memory Card Status

The access to the memory card is indicated by the SD-LED.

The SD-LED is directly above the memory card slot behind the transparent cover flap.

11.1.4 Evaluating Power Supply Status

The power supply unit of the device has two green LEDs that indicate the status of the power supplies.

LED “A” indicates the 24 V supply of the fieldbus coupler.

LED “B” or “C” reports the power available on the power jumper contacts for field side power.

Table 94: Power Supply Status Diagnostics – Solution in Event of Error

LED Status	Meaning	Solution
A		
Green	Operating voltage for the system is available.	-
Off	No power is available for the system	Check the power supply for the system (24 V and 0 V).
B or C		
Green	The operating voltage for power jumper contacts is available.	-
Off	No operating voltage is available for the power jumper contacts.	Check the power supply for the power jumper contacts (24 V and 0 V).

11.2 Fault Behavior

11.2.1 Loss of Fieldbus

A fieldbus and, hence, a link failure is recognized when the set reaction time for the watchdog expires without initiation by the higher-order control system. This may occur, for example, when the Master is switched off, or when there is a disruption in the bus cable. An error at the Master can also result in a fieldbus failure. No connection via ETHERNET.

The MODBUS watchdog monitors the ongoing MODBUS communication via MODBUS protocol. A fieldbus failure is signaled by the red "I/O" LED lighting up, provided the MODBUS watchdog has been configured and activated.

Fieldbus monitoring independently of a certain protocol is possible using the function block 'FBUS_ERROR_INFORMATION' in the library "Mod_com.lib". This checks the physical connection between modules and the controller and assumes evaluation of the watchdog register in the control system program. The I/O bus remains operational and the process images are retained. The control system program can also be processed independently.

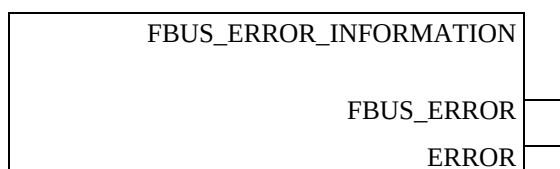


Figure 94: Function Block for Determining Loss of Fieldbus, Independently of Protocol

'FBUS_ERROR' (BOOL)	= FALSE	= no fault
	= TRUE	= loss of field bus
'ERROR' (WORD)	= 0	= no fault
	= 1	= loss of field bus

The node can be put into a safe status in the event of a fieldbus failure with the aid of these function block outputs and an appropriately programmed control system program.



Information

Loss of fieldbus detection through MODBUS protocol:

For detailed information about the watchdog register, refer to Section “MODBUS Functions”, in particular Section “Watchdog (Fieldbus failure)”.

Protocol-independent detection of loss of fieldbus:

The library “Mod_com.lib” with function block ‘FBUS_ERROR_INFORMATION’ is normally included in the setup for the WAGO-I/O-PRO. You can integrate the library via register “Resources” at the bottom on the left of the workspace. Click **Insert** and then **Other libraries**. The “Mod_com.lib” is located in folder C:\Programme\WAGO Software\CODESYS V2.3\Targets\WAGO\Libraries\32_Bit

11.2.2 Internal Data Bus Failure

I/O LED indicates an internal bus failure.

I/O LED flashed red:

When an internal data bus failure occurs, the fieldbus controller generates an error message (error code and error argument).

An internal data bus failure occurs, for example, if an I/O module is removed.

If the error occurs during operation, the output modules operate as they do during an internal data bus stop.

If the internal data bus error is resolved, the fieldbus controller starts up after turning the power off and on similar to that of a normal start-up. The process data is transmitted again and the outputs of the node are set accordingly.

If the “KBUS_ERROR_INFORMATION” function block is evaluated in the control program, then the “ERROR”, “BITLEN”, “TERMINALS” and “FAILADDRESS” output values are relevant.

ERROR	= FALSE	= No fault
(BITLEN		= Bit length of the internal bus shift register
TERMINALS		= Number of I/O modules)
ERROR	= TRUE	= Internal Bus Error
(BITLEN		= 0
TERMINALS		= 0
FAILADDRESS		= Position of the I/O module after which the internal bus interruption arose, similar to the flashed error argument of the I/O LED)

12 Fieldbus Communication

Fieldbus communication between master application and a WAGO fieldbus coupler/controller based on the ETHERNET standard normally occurs via an implemented fieldbus-specific application protocol.

Depending on the application, this can be e.g., MODBUS/TCP (UDP), EtherNet/IP, BACnet/IP, KNXnet/IP, PROFINET, sercos or other.

In addition to the ETHERNET standard and the fieldbus-specific application protocol, there are also other communications protocols important for reliable communication and data transmission and other related protocols for configuring and diagnosing the system implemented in the WAGO fieldbus coupler/controller based on ETHERNET.

These protocols are explained in more detail in the other sections.

12.1 Implemented Protocols

12.1.1 Communication Protocols

12.1.1.1 IP (Internet Protocol)

The Internet protocol divides datagrams into segments and is responsible for their transmission from one network subscriber to another. The stations involved may be connected to the same network or to different physical networks which are linked together by routers.

Routers are able to select various paths (network transmission paths) through connected networks, and bypass congestion and individual network failures. However, as individual paths may be selected which are shorter than other paths, datagrams may overtake each other, causing the sequence of the data packets to be incorrect.

Therefore, it is necessary to use a higher-level protocol, for example, TCP to guarantee correct transmission.

IP Packet

In addition to the data units to be transported, the IP data packets contain a range of address information and additional information in the packet header.

Table 95: IP Packet

IP Header	IP Data

The most important information in the IP header is the IP address of the transmitter and the receiver and the transport protocol used.

IP Addresses

To allow communication over the network each fieldbus node requires a 32 bit Internet address (IP address).



Note

IP Address must be unique!

For error free operation, the IP address must be unique within the network. The same IP address may not be assigned twice.

As shown below there are various address classes with net identification (net ID) and subscriber identification (subscriber ID) of varying lengths. The net ID defines the network in which the subscriber is located. The subscriber ID identifies a particular subscriber within this network.

Networks are divided into various network classes for addressing purposes:

- **Class A:** (Net ID: Byte 1, Host ID: Byte 2... Byte 4)

Table 96: Network Class A

e. g.	101	.	16	.	232	.	22
	01100101		00010000		11101000		00010110
0	Net ID		Host ID				

The highest bit in Class A networks is always '0'. This means the highest byte can be in a range of '0 0000000' to '0 1111111'.

Therefore, the address range of a Class A network in the first byte is always between 0 and 127.

- **Class B:** (Net ID: Byte 1 ... Byte 2, Host ID: Byte 3... Byte 4)

Table 97: Network Class B

e. g.	181	.	16	.	232	.	22
	10110101		00010000		11101000		00010110
10	Net ID		Host ID				

The highest bits in Class B networks are always '10'. This means the highest byte can be in a range of '10 000000' to '10 111111'.

Therefore, the address range of Class B networks in the first byte is always between 128 and 191.

- **Class C:** (Net ID: Byte 1 ... Byte 3, Host ID: Byte 4)

Table 98: Network Class C

e. g.	201	.	16	.	232	.	22
	11000101		00010000		11101000		00010110
110	Net ID		Host ID				

The highest bits in Class C networks are always '110'. This means the highest byte can be in a range of '110 00000' to '110 11111'.

Therefore, the address range of Class C networks in the first byte is always between 192 and 223.

- **Additional network classes (D, E):** are only used for special tasks.

Key Data

Table 99: Key Data Class A, B and C

Network Class	Address range of the subnetwork	Possible number of	
		Networks	Hosts per Network
Class A	0.XXX.XXX.XXX ... 127.XXX.XXX.XXX	128 (2^7)	Approx. 16 Million (2^{24})
Class B	128.000.XXX.XXX ... 191.255.XXX.XXX	Approx. 16 Thousand (2^{14})	Ca. 65 Thousand (2^{16})
Class C	192.000.000.XXX ... 223.255.255.XXX	Approx. 2 Million (2^{21})	254 (2^8)

Each WAGO ETHERNET fieldbus coupler or controller can be easily assigned an IP address via the implemented BootP protocol. For small internal networks we recommend selecting a network address from Class C.

Note



Do not set IP addresses to 0.0.0.0 or 255.255.255.255!

Never set all bits to equal 0 or 1 in one byte (byte = 0 or 255). These are reserved for special functions and may not be allocated. Therefore, the address 10.0.10.10 may not be used due to the 0 in the second byte.

If a network is to be directly connected to the Internet, only registered, internationally unique IP addresses allocated by a central registration service may be used. These are available from InterNIC (International Network Information Center).

Note



Internet access only by the authorized network administrator!

Direct connection to the Internet should only be performed by an authorized network administrator and is therefore not described in this manual.

Subnets

To allow routing within large networks a convention was introduced in the specification RFC 950. Part of the Internet address, the subscriber ID is divided up again into a subnetwork number and the station number of the node. With the aid of the network number it is possible to branch into internal subnetworks within the partial network, but the entire network is physically connected together. The size and position of the subnetwork ID are not defined; however, the size is dependent upon the number of subnets to be addressed and the number of subscribers per subnet.

Table 100: Example: Class B Address with Field for Subnet IDs

1	8	16	24	32
1	0	...	Network ID	Subnet ID
Host ID				

Subnet Mask

A subnet mask was introduced to encode the subnets in the Internet. This involves a bit mask, which is used to mask out or select specific bits of the IP address. The mask defines the subscriber ID bits used for subnet coding, which denote the ID of the subscriber. The entire IP address range theoretically lies between 0.0.0.0 and 255.255.255.255. Each 0 and 255 from the IP address range are reserved for the subnet mask.

The standard masks depending upon the respective network class are as follows:

- Class A Subnet mask:**

Table 101: Subnet Mask for Class A Network

255	.0	.0	.0

- Class B Subnet mask:**

Table 102: Subnet Mask for Class B Network

255	.255	.0	.0

- Class C Subnet mask:**

Table 103: Subnet Mask for Class C Network

255	.255	.255	.0

Depending on the subnet division the subnet masks may, however, contain other values beyond 0 and 255, such as 255.255.255.128 or 255.255.255.248.

Your network administrator allocates the subnet mask number to you.

Together with the IP address, this number determines which network your PC and your node belongs to.

The recipient node, which is located on a subnet, initially calculates the correct network number from its own IP address and subnet mask. Only then the node checks the node number and, if it corresponds, delivers the entire packet frame.

Table 104: Example for an IP Address from a Class B Network

IP address	172.16.233.200	'10101100 00010000 11101001 11001000'
Subnet mask	255.255.255.128	'11111111 11111111 11111111 10000000'
Net ID	172.16.0.0	'10101100 00010000 00000000 00000000'
Subnet ID	0.0.233.128	'00000000 00000000 11101001 10000000'
Host ID	0.0.0.72	'00000000 00000000 00000000 01001000'



Note

Specification of the network mask necessary!

Specify the network mask defined by the administrator in the same way as the IP address when installing the network protocol.

Gateway

The subnets of the Internet are normally connected via gateways. The function of these gateways is to forward packets to other networks or subnets.

This means that in addition to the IP address and network mask for each network card, it is necessary to specify the correct IP address of the standard gateway for a PC or fieldbus node connected to the Internet. You should also be able to obtain this IP address from your network administrator.

The IP function is limited to the local subnet if this address is not specified.

To communicate directly with each other, host and gateway must be on the same subnet, that means the network ID must be the same.

RAW IP

Raw IP manages without protocols such as PPP (point-to-point protocol). With RAW IP, the TCP/IP packets are directly exchanged without handshaking, thus enabling the connection to be established more quickly.

However, the connection must beforehand have been configured with a fixed IP address. The advantages of RAW IP are high data transfer rate and good stability.

IP Multicast

Multicast refers to a method of transmission from a point to a group, which is a point-to-multipoint transfer or multipoint connection. The advantage of multicast is that messages are simultaneously transferred to several users or closed user groups via one address.

IP multicasting at the Internet level is realized with the help of the Internet Group Message Protocol IGMP; neighboring routers use this protocol to inform each other on membership to the group.

For distribution of multicast packets in the sub-network, IP assumes that the datalink layer supports multicasting. In the case of Ethernet, you can provide a packet with a multicast address in order to send the packet to several recipients with a single send operation. Here, the common medium enables packets to be sent simultaneously to several recipients. The stations do not have to inform each other on who belongs to a specific multicast address – every station physically receives every packet. The resolution of IP address to Ethernet address is solved

by the use of algorithms, IP multicast addresses are embedded in Ethernet multicast addresses.

12.1.1.2 TCP (Transmission Control Protocol)

As the layer above the Internet protocol, TCP (Transmission Control Protocol) guarantees the secure transport of data through the network.

TCP enables two subscribers to establish a connection for the duration of the data transmission. Communication takes place in full-duplex mode (i.e., transmission between two subscribers in both directions simultaneously).

TCP provides the transmitted message with a 16-bit checksum and each data packet with a sequence number.

The receiver checks that the packet has been correctly received on the basis of the checksum and then sets off the sequence number. The result is known as the acknowledgement number and is returned with the next self-sent packet as an acknowledgement.

This ensures that the lost TCP packets are detected and resent, if necessary, in the correct sequence.

TCP Data Packet

The packet header of a TCP data packet is comprised of at least 20 bytes and contains, among others, the application port number of the transmitter and the receiver, the sequence number and the acknowledgement number.

The resulting TCP packet is used in the data unit area of an IP packet to create a TCP/IP packet.

TCP Port Numbers

TCP can, in addition to the IP address (network and subscriber address), respond to a specific application (service) on the addressed subscriber. For this the applications located on a subscriber, such as a web server, FTP server and others are addressed via different port numbers. Well-known applications are assigned fixed ports to which each application can refer when a connection is built up (Examples: Telnet Port number: 23, http Port number: 80).

A complete list of “standardized services” is contained in the RFC 1700 (1994) specifications.

12.1.1.3 UDP (User Datagram Protocol)

The UDP protocol, like the TCP protocol, is responsible for the transport of data. Unlike the TCP protocol, UDP is not connection-orientated; meaning that there are no control mechanisms for the data exchange between transmitter and receiver. The advantage of this protocol is the efficiency of the transmitted data and the resulting higher processing speed.

12.1.2 Configuration and Diagnostics Protocols

12.1.2.1 BootP (Bootstrap Protocol)

The “Bootstrap Protocol” (BootP) can be used to assign an IP address and other parameters to the fieldbus coupler/controller in a TCP/IP network. Subnet masks and gateways can also be transferred using this protocol. Protocol communication is comprised of a client request from the fieldbus coupler or controller and a server response from the PC.

A broadcast request is transmitted to Port 67 (BootP server) via the protocol that contains the hardware address (MAC ID) for the fieldbus coupler or controller.

The BootP server then receives this message. The server contains a database in which the MAC ID and IP addresses are assigned to one another. When a MAC address is found a broadcast reply is transmitted via the network.

The fieldbus coupler/controller “listens” at the specified Port 68 for a response from the BootP server. Incoming packets contain information such as the IP address and the MAC address for the fieldbus coupler/controller. A fieldbus coupler/controller recognizes by the MAC address that the message is intended for that particular fieldbus coupler/controller and accepts the transmitted IP address into its network.

Note



IP addresses can be assigned via BootP under Windows and Linux!

You can use WAGO-BootP-Server to assign an IP address under the Windows and Linux operating systems. You can also use any other BootP server besides WAGO-BootP-Server. You can also use any other BootP server besides the WAGO-BootP-Server.

Information



More information about WAGO-BootP-Server

The process for assigning addresses using WAGO-BootP-Server is described in detail in the section “Commissioning”.

The BootP Client assists in the dynamic configuration of the network parameters: The ETHERNET TCP/IP fieldbus controller has a BootP client that supports the following options in addition to the default “IP address” option:

Table 105: BootP Options

Option	Meaning
[OPT1] Subnet mask	32-bit address mask that displays which bits of the IP address identify the network and which identify the network stations.
[OPT2] Time zone	Time difference between the local time and the UTC (Universal Time Coordinated).
[OPT3] Gateway	IP address of the router that permits access to other networks.
[OPT6] DNS server	IP address of the name servers that converts a name into an IP address. Up to 2 DNS servers can be configured.
[OPT12] Host name	The name of the host is the unique name of a computer in a network. The host name can contain up to 32 characters.
[OPT15] Domain name	The name of the domain is the unique name of a network. The domain name can contain up to 32 characters.
[OPT42] NTP server	IP address of the Network Time Server. When assigning an NTP server, the SNTP client is automatically enabled in the coupler.

The “Features” WBM page can also be used to select the “BootP Request before static IP” option. After the restart, 5 BootP queries are sent. If there is no response to any of these queries, the fieldbus coupler/controller tries to configure itself with the IP parameters saved in the EEPROM.

If you want the IP address and subnet mask are stored in the EEPROM when using the Bootstrap protocol, then the option “use IP from EEPROM” has to be switched (via the WBM, HTML page “Port”) following the configuration via BootP.

When booting next the fieldbus coupler/controller uses the parameters saved in the EEPROM.

By default, BootP is activated in the fieldbus coupler/controller.

When BootP is activated, the fieldbus coupler/controller expects the BootP server to be permanently available.

If there is no BootP server available after a PowerOn reset, the network will remain inactive.

If there is an error in the saved parameters, the I/O LED reports a blink code and configuration via BootP is turned on automatically.

12.1.2.2 DHCP (Dynamic Host Configuration Protocol)

The fieldbus coupler/controller internal HTML page opened via the “Port” link provides the option to configure the network using the data saved in the EEPROM or via DHCP instead of via the BootP protocol.

DHCP (Dynamic Host Configuration Protocol) is a further development of BootP and is backwards compatible with BootP.

Both BOOTP and DHCP assign an IP address to the fieldbus node (Client) when starting; the sequence is the same as for BootP.

For configuration of the network parameters via DHCP, the fieldbus coupler/controller sends a client request to the DHCP server e.g., on the connected PC.

A broadcast request is transmitted to Port 67 (DHCP server) via the protocol that contains the hardware address (MAC ID) for the fieldbus coupler/controller.

The DHCP server then receives this message. The server contains a database in which the MAC ID and IP addresses are assigned to one another. When a MAC address is found a broadcast reply is transmitted via the network.

The fieldbus coupler/controller “listens” at the specified Port 68 for a response from the DHCP server. Incoming packets contain information such as the IP address and the MAC address for the fieldbus coupler/controller. A fieldbus coupler/controller recognizes by the MAC address that the message is intended for that particular fieldbus coupler/controller and accepts the transmitted IP address into its network.

If there is no reply, the inquiry is sent again after 4 seconds, 8 seconds and 16 seconds.

If all inquiries receive no reply, a blink code is reported via the I/O LED.

If you want the IP address and subnet mask are stored in the EEPROM when using DHCP, then the option “use IP from EEPROM” has to be switched (via the WBM, HTML page “Port”) following the configuration via DHCP.

When booting next the fieldbus coupler/controller uses the parameters saved in the EEPROM.

The difference between BOOTP and DHCP is that both use different assignment methods and that configuration with DHCP is time limited. The DHCP client always has to update the configuration after the time has elapsed. Normally, the same parameters are continuously confirmed by the server.

BOOTP can be used to assign a fixed IP address for each client where the addresses and their reservation are permanently saved in the BOOTP server database.

Because of this time dependency, DHCP is also used to dynamically assign available IP addresses through client leases (lease time after which the client requests a new address) where each DHCP client address is saved temporarily in the server database.

In addition, DHCP clients do not require a system restart to rebind or renew configuration with the DHCP server. Instead, clients automatically enter a rebinding state at set timed intervals to renew their leased address allocation with the DHCP server. This process occurs in the background and is transparent to the user.

There are three different operating modes for a DHCP server:

- **Manual assignment**

In this mode, the IP addresses are permanently assigned on the DHCP server to specific MAC addresses. The addresses are assigned to the MAC address for an indefinite period.

Manual assignments are used primarily to ensure that the DHCP client can be reached under a fixed IP address.

- **Automatic assignment**
For automatic assignment, a range of IP addresses is assigned on the DHCP server.
If the address was assigned from this range once to a DHCP client, then it belongs to the client for an indefinite period as the assigned IP address is also bound to the MAC address.
- **Dynamic assignment**
This process is similar to automatic assignment, but the DHCP server has a statement in its configuration file that specifies how long a certain IP address may be “leased” to a client before the client must log into the server again and request an “extension”.
If the client does not log in, the address is released and can be reassigned to another (or the same) client. The time defined by the administrator is called Lease Time.
Some DHCP servers also assign IP addresses based on the MAC address, i.e., a client receives the same IP address as before after longer network absence and elapse of the Lease Time (unless the IP address has been assigned otherwise in the mean time).

DHCP is used to dynamically configure the network parameters.
The ETHERNET TCP/IP fieldbus controller has a DHCP client that supports the following options in addition to the default “IP address” option:

Table 106: Meaning of DHCP Options

Option	Meaning
[OPT1] Subnet mask	32-bit address mask that displays which bits of the IP address identify the network and which identify the network stations.
[OPT2] Time zone	Time difference between the local time and the UTC (Universal Time Coordinated).
[OPT3] Gateway	IP address of the router that permits access to other networks.
[OPT6] DNS server	IP address of the name servers that converts a name into an IP address. Up to 2 DNS servers can be configured.
[OPT15] Domain name *)	The name of the domain is the unique name of a network. The domain name can contain up to 32 characters.
[OPT42] NTP server	IP address of the Network Time Server. When assigning an NTP server, the SNTP client is automatically enabled in the coupler.
[OPT51] Lease time	The maximum duration (i.e., how long the fieldbus coupler/controller maintains the assigned IP address) can be defined here. The maximum lease time for the fieldbus controller is 48 days. This is due to the internal timer resolution.
[OPT58] Renewing time	The renewing time indicates when the fieldbus coupler/controller must renew the lease time. The renewing time should be approximately half of the lease time.
[OPT59] Rebinding time	The rebinding time indicates after what amount of time the fieldbus coupler/controller must have received its new address. The rebinding time should be approximately 7/8 of the lease time.

*) In contrast to BootP, the DHCP client does not support assignment of the host name.

12.1.2.3 HTTP (Hypertext Transfer Protocol)

HTTP is a protocol used by WWW (World Wide Web) servers for the forwarding of hypermedia, texts, images, audiodata, etc.

Today, HTTP forms the basis of the Internet and is also based on requests and responses in the same way as the BootP protocol.

The HTTP server implemented in the (programmable) fieldbus coupler or controller is used for viewing the HTML pages saved in the coupler/controller. The HTML pages provide information about the coupler/controller (state, configuration), the network and the process image.

On some HTML pages, (programmable) fieldbus coupler or controller settings can also be defined and altered via the web-based management system (e.g. whether IP configuration of the coupler/controller is to be performed via the DHCP protocol, the BootP protocol or from the data stored in the EEPROM).

The HTTP server uses port number 80.

12.1.2.4 DNS (Domain Name Systems)

The DNS client enables conversion of logical Internet names such as www.wago.com into the appropriate decimal IP address represented with separator stops, via a DNS server. Reverse conversion is also possible.

The addresses of the DNS server are configured via DHCP, BootP or web-based management. Up to 2 DNS servers can be specified. The host identification can be achieved with two functions; an internal host table is not supported.

12.1.2.5 SNTP-Client (Simple Network Time Protocol)

The SNTP client is used for synchronization of the time of day between a time server (NTP and SNTP server Version 3 and 4 are supported) and the internal system time in the (programmable) fieldbus coupler or controller. The protocol is executed via a UDP port. Only unicast addressing is supported.

Configuration of the SNTP client

The configuration of the SNTP client is performed via the web-based management system. The following parameters must be set:

Table 107: Meaning of the SNTP Parameters

Parameter	Meaning
Address of the Time Server	The address assignment can be made either over an IP address or a hostname.
WBM page "TCP/IP" → „SNTP Update Time (sec, max. 65535) ”	The update time indicates the interval in seconds, in which the synchronization with the time server is to take place.
WBM page "Clock" → "Timezone (+/- hour:minute)"	The time zone relative to GMT (Greenwich Mean time). A range of -12 to +14 hours is acceptable.

WBM page “Port” → “SNTP”	It indicates whether the SNTP Client is to be activated or deactivated.

12.1.2.6 FTP-Server (File Transfer Protocol)

The file transfer protocol (FTP) enables files to be exchanged between different network stations regardless of operating system.

In the case of the ETHERNET coupler/controller, FTP is used to store and read the HTML pages created by the user, the IEC61131 program and the IEC61131 source code in the (programmable) fieldbus coupler or controller.

A total memory of 4,5 MB is available for the file system.



Note

Cycles for flash limited to 1 million!

Up to 1 million write cycles per sector are allowed when writing the flash for the file system. The file system supports “Wear-Leveling”, so that the same sectors are not always written to.

The following table shows the supported FTP commands for accesses to the file system:

Table 108: FTP Commands and Function

Command	Function
USER	Identification of the user
PASS	User password
ACCT	Account for access to certain files
REIN	Server reset
QUIT	Terminates the connection
PORT	Addressing of the data link
PASV	Changes server in the listen mode
TYPE	Determines the kind of the representation for the transferred file
STRU	Determines the structure for the transferred file
MODE	Determines the kind of file transmission
RETR	Reads file from server
STOR	Saves file on server
APPE	Saves file on server (Append mode)
ALLO	Reservation of the necessary storage location for the file
RNFR	Renames file from (with RNTD)
RNTD	Renames file in (with RNFR)
ABOR	Stops current function
DELE	Deletes file
CWD	Changes directory
LIST	Gives the directory list
NLST	Gives the directory list
RMD	Deletes directory
PWD	Gives the actually path
MKD	Creates a directory

The TFTP (Trivial File Transfer Protocol) is not supported by some of the couplers/controllers.



Information

More Information about the implemented Protocols

You can find a list of the exact available implemented protocols in the section “Technical Data” to the fieldbus coupler and/or controller.

12.1.2.7 SNMP (Simple Network Management Protocol)

The Simple Network Management Protocol (SNMP) is responsible for transporting the control data that allows the exchange of management information as well as status and statistic data between individual network components and a management system.

An SNMP management workstation polls the SNMP agents to obtain information on the relevant devices.

SNMP is supported in versions 1/2c and some fieldbus couplers/controllers in version 3.

This represents a community message exchange in SNMP versions 1 and 2c. The community name of the network community must thereby be specified.

In SNMP version 3, exchanging messages is user-related. Each device, that knows the passwords set via WBM, may read or write values from the controller. In SNMPv3, user data from SNMP messages can also be transmitted in encoded form. This way, both requested values and values to be written cannot be easily decoded by others via ETHERNET. This is why SNMPv3 is often used in safety-related networks.

The device data, that can be accessed or modified by an SNMP agent, is called SNMP object. The sets of SNMP objects are stored in a logical database called Management Information Base (MIB); this is why these objects are typically known as “MIB objects”.

The SNMP of the ETHERNET controller includes both the general MIB acc. to RFC1213 (MIB II) and a special WAGO MIB.

SNMP is processed via port 161. The port number for SNMP traps (agent messages) is 161. Both ports must be enabled to use SNMP.

12.1.2.7.1 MIB II Description

MIB II acc. to RFC1213 is divided into the following groups:

Table 109: MIB II Groups

Group	Identifier
System Group	1.3.6.1.2.1.1
Interface Group	1.3.6.1.2.1.2
IP Group	1.3.6.1.2.1.4
IpRoute Table Group	1.3.6.1.2.1.4.21
ICMP Group	1.3.6.1.2.1.5
TCP Group	1.3.6.1.2.1.6
UDP Group	1.3.6.1.2.1.7
SNMP Group	1.3.6.1.2.1.11



Information

Additional Information

Please find detailed information on these individual groups in section "MIB II groups" of the manual appendix.

12.1.2.7.2 Traps

Standard Traps

For specific events, the SNMP agent will independently send one of the following messages without polling the manager.



Note

Enable event messages (traps) in the WBM!

Initially enable the event messages in the WBM in menu "SNMP" under "Trap Enable" by enabling "V1".

The following messages are triggered automatically as traps (SNMPv1) by the fieldbus coupler/controller:

Table 110: Standard Traps

TrapType/TrapNumber/ OID of the provided value	Name	Event
TrapType = 0	ColdStart	Restart the coupler/controller
TrapType = 1	WarmStart	Reset via service switch
TrapType = 3	EthernetUp	Network connection detected
TrapType = 4	AuthenticationFailure	Unauthorized (abortive) MIB access
TrapType = 6/ ab Trap-Nummer 25 benutzerspezifisch	enterpriseSpecific	Enterprise-specific messages and function poll in the PFC program starting with enterprise trap number 25

12.1.3 Application Protocols

If fieldbus specific application protocols are implemented, then the appropriate fieldbus specific communication is possible with the respective coupler/controller. Thus the user is able to have a simple access from the respective fieldbus on the fieldbus node.

The implemented fieldbus specific application protocols these protocols are individual described in the following chapters.

12.1.3.1 BACnet/IP

The "Building Automation and Control Network", BACnet for short, is a standardized and company-neutral network protocol for building automation, and is originally geared towards the area of heating, ventilation and air conditioning (HVAC).

This protocol has been an ASHRAE standard since 1995, was accepted as a standard by ANSI (ANSI/ASHRAE 135-2004) and has become anchored in the DIN EN ISO standard 16484-5 (Building Automation Systems, Data Communication Protocols). The following chapter refers to the content of the BACnet standard ANSI/ASHRAE 135-2010, DIN EN ISO 16484-5.

The BACnet network model works independently of the system. When using devices, data/information is represented by predefined objects and transmitted with the aid of services over a special BACnet network layer. Communication is based on the client/server method.

Along with objects and services, network technologies of the lowest levels are also specified in the Standard.

12.1.3.1.1 Interoperability

The objective of the BACnet Standard is the creation of an open, interoperable network. The requirement for the cooperation of different participants in a BACnet network is a common understanding of the BACnet Standard and therefore of the use of common communication structures.

For this purpose, each device has a Protocol Implementation Conformance Statement - PICS). The PICS is a document containing objects, coding, routing information and BACnet Interoperability Building Blocks - BIBBs defined by the Standard that the end device supports. These BIBBs indicate which services the client and server must support in order to fulfill certain system requirements.

Through the common understanding by customers and manufacturers of the protocol standard and common functions and communication structures in the PICS, the interoperability of the network is ensured.

12.1.3.1.2 BACnet Components

According to the BACnet Standard, 50 different objects and 37 services are supported. An object is composed of several object-specific properties. Services are also present and transmit their data in certain structures.

12.1.3.1.2.1 Objects

BACnet offers a unified structure for different areas in the network. Thus, field devices such as probes, controllers of the automation level or even complex control and operating stations of the management level are modeled in the object representation. The 50 objects are tailored specifically to the HVAC sector for building automation services. They contain both physical inputs and outputs and virtual objects.

Table 111: BACnet Objects

No	Object	Function
0	Analog Input	Analog inputs, measures voltage, temperature, etc.
1	Analog Output	Analog outputs, e.g. for setting controllers
2	Analog Value	Analog value; e.g. from a calculation
3	Binary Input	Binary input, reports an event or disturbance
4	Binary Output	Binary output; alternates between two states
5	Binary Value	Binary value, virtual binary data time
6	Calendar	Operating calendar, list of holidays and vacations
7	Command	(Group) request to execute predefined activities (e.g. a list of switching commands)
8	Device	Device, information on the respective BACnet device
9	Event Enrollment	Event category, defined reactions to events
10	File	File, transmits data
11	Group	Group entry, grouped object values in the device
12	Loop	Controller, performs regulation functions
13	Multi State Input	Multi-level input, delivers reports on states such as off/on, open/closed as a coded number
14	Multi State Output	Multi-level output, delivers the output states of commands
15	Notification Class	Notification class, assigns alarm and event reports to times and recipients
16	Program	Program, used to access programs in the BACnet device
17	Schedule	Schedule, for establishing certain actions at predefined times
18	Averaging	Average for calculations and statistics
19	Multi State Value	Multi-level value, delivers logical states of the object
20	Trend Log	Trend log of properties of certain criteria, either cyclic or after value changes (COV)
21	Life Safety Point	Hazard reporter, delivers information on properties with regard to notifications of danger
22	Life Safety Zone	Security area, includes hazard reporting objects according to certain criteria
23	Accumulator	Count value entry; counts values
24	Pulse Converter	Impulse entry, counts quantities for control and monitoring
25	Event-Log	Objects for transmitting event-related values and time stamp
26	Global-Group	Global group inputs, object for grouping readable properties of any object in the entire network
27	Trend-Log-Multiple	Object for trend recording for several values, also device- and network-wide

Table 111: BACnet Objects

No	Object	Function
28	Load-Control	Objects for maximum load limit, load control
29	Structured-View	Object to support the representation of object relationships to one another, tree view
30	Access-Door	Access control / evaluation of door components, e.g. contacts
32	Access-Credential	Access control data such as cards or PIN
33	Access-Point	Control point for access check
34	Access-Rights	Access rights and conditions
35	Access-User	Persons/groups to which user permissions are assigned
36	Access-Zone	Secure access area
37	Credential-Data-Input	Input data for access permissions (e.g. for card reader or PIN input devices)
38	Network-Security	Represents network security settings of a BACnet device
39	BitString-Value	Represents a bit string
40	CharacterString-Value	Represents a character string
41	Date-Pattern-Value	Represents recurring date information
42	Data-Value	Date display
43	DateTime-Pattern-Value	Represents recurring date/time combinations
44	DateTime-Value	Represents a date/time combination
45	Integer-Value	Display of positive and negative integer values
46	Large-Analog-Value	Depiction of double-precision floating-point numbers
47	OctetString-Value	Depiction of a character string as an octet (8 bit)
48	Positive-Integer-Value	Display of positive integer values
49	Time-Pattern-Value	Represents recurring times
50	Time-Value	Depiction of a time

12.1.3.1.2.2 Properties

Objects are described by specific properties with their values. In this manner, object information, such as name, status and behavior of the object in question can be read.

The following three properties are common to all objects:

- Object_Identifier
- Object_Name
- Object_Type

The values of the properties are visible and readable (**R**) throughout the entire BACnet network. Some are also writable (**W**) by remote BACnet devices, depending on property and configuration.

Object properties are accessed through services.

12.1.3.1.2.3 Services

BACnet is based on the Client-Server model. The Client makes requests to the Server, which processes the Client's requests and returns a report.

37 standardized services are divided into 5 categories (ANSI/AHRAE Standard 135-2010)

- Alarm and Event Services
- Object Access Services
- File Access Services
- Remote Device Management Service
- Virtual Terminal Service

Alarm and Event Services

Responses to alarm and event reports can be one of three types

1. Change of Value Reporting (COV)

Data are only sent if they have been altered and the Client is registered for COV Reporting in a COV Server.

2. Intrinsic Reporting

To send alarm or event reports to one or more recipients, the notification and notification class are first sent to the Notification Class Object. This object contains a list of those recipients that are registered for this notification class and sends the report to these recipients.

3. Algorithmic Change Reporting

Algorithmic Change Reporting allows a BACnet device to have alarm and event notifications available that are defined by the Event Enrollment Object. It can be set for every property of every object. The same algorithms apply as for Intrinsic Reporting, CHANGE_OF_BITSTRING, CHANGE_OF_STATE, CHANGE_OF_VALUE, COMMAND_FAILURE, FLOATING_LIMIT, OUT_OF_RANGE, BEFFER_READY, CHANGE_OF_LIFE_SAFETY, UNSIGNED_RANGE, ACCESS_EVENT, CHANGE_OF_CHARACTERSTRING, DOUBLE_OUT_OF_RANGE, SIGNED_OUT_OF_RANGE, UNSIGNED_OUT_OF_RANGE, CHANGE_OF_STATUS_FLAGS

The listed mechanisms can be used for the following alarm and event services:

- **AcknowledgeAlarm Service**
To confirm an alarm report, it may be necessary for a person to implement and confirm an alarm.
- **ConfirmedCOVNotification Service**
Informs subscribers of a change in property in a certain object. A confirmation is expected.
- **UnconfirmedCOVNotification Service**
Informs subscribers of changes in properties of a certain object. No confirmation is expected.
- **ConfirmedEventNotification Service**
Informs of an event that has occurred and requests a confirmation.

- **UnconfirmedEventNotification Service**
Informs of an event that has occurred, but does not request a confirmation (broadcast, multicast).
- **GetAlarmSummary Service**
Queries available alarms of a device. The Event_State property of the object cannot be NORMAL in this case. The Notify_Type must be set to ALARM.
- **GetEnrollmentSummary Service**
Queries a list of objects that send event notifications. These can be filtered according to confirmation (obligatory), events, priorities, etc.
- **GetEventInformation Service**
Queries available event states of a device. The Event_State property of the object cannot be NORMAL in this case. In the case of the Acked_Transitions property, one of the three bits (TO-OFFNORMAL, TO-FAULT, TO-NORMAL) must be FALSE.
- **LifeSafetyOperation Service**
Transmits specific instructions from the user for fire, lifesaving and security systems, e.g. for switching acoustic and/or optical signals off or on.
- **SubscribeCOV Service**
Request for acceptance into the recipient list for changes of value.
- **SubscribeCOVProperty Service**
Request for acceptance into the recipient list for changes of value in certain properties.

Object Access Services

- **AddListElement Service**
Adds one or more list elements to object properties consisting of lists.
- **RemoveListElement Service**
Deletes one or more elements from lists, even if these are lists themselves.
- **CreateObject Service**
A new object instance is generated in the server. This is reinitialized by either the CreateObject Service or the WriteProperty Service.
- **DeleteObject Service**
Deletes selected objects.
- **ReadProperty Service**
Reads the value of an object property
- **ReadPropertyMultiple Service**
Reads the content of several object properties.

- **ReadRange Service**
Reads the data range of an object property.
- **WriteProperty-Service**
Writes a value into a specific property of an object.
- **WritePropertyMultiple-Service**
Changes the values of several properties of several objects.

File Access Services

- **AtomicReadFile Service**
Query of a file, which is opened, read and closed again, as a whole or in part.
- **AtomicWriteFile Service**
Query of a file, which is opened, read and closed again, as a whole or in part.

Remote Device Management Service

- **DeviceCommunicationControl Service**
Clients instruct remote devices to stop communication with all protocol units in the application layer (Application Protocol Data Units - APDUs) for a certain time. This function is mostly used for diagnostic purposes. Clients need a password for this service.
- **ConfirmedPrivateTransfer Service**
Allows a BACnet Client to call up non-standardized services; a manufacturer identification code (issued by ASHRAE) as well as a service number must be given for these. The service is confirmed.
- **UnconfirmedPrivateTransfer Service**
Allows a BACnet Client to call up non-standardized services; a manufacturer identification code and a service number must be given for these. The service is not confirmed.
- **ReinitializeDevice Service**
The Client can trigger a restart in a remote device or monitor backup and restore settings there.
- **ConfirmedTextMessage Service**
Sending of text messages, which are confirmed; can be classified as "normal" and "urgent"; no support of broadcast and multicast.
- **UnconfirmedTextMessage Service**
Sending of text messages; can be classified as "normal" and "urgent"; no support of broadcast and multicast.

-
- **TimeSynchronization Service**
Synchronization of time
 - **UTCTimeSynchronization Service**
Synchronization of time to Universal Time Coordinated (UTC)
 - **Who-Has and I-Have Services**
Search for a certain Object_Name or Object-Identifier or its reference in other devices / response to this request
 - **Who-Is and I-Am Services**
Query Object_Identifier and/or network address of other network members / respond to a request of other participants

Virtual Terminal Service

- **VT-Open Service**
Establishes a connection to a network.
- **VT-Close Service**
Breaks the connection off.
- **VT-Data Service**
Transmits data between network members.

12.1.3.1.2.3.1 Client-Server Communication

In Client-Server communication, Clients and Servers are considered to have different functions. A Client makes a request as a service user of a Server (service provider). The Server then sends its response back to the Client.

BACnet devices can work, depending on function, both as a Client and as a Server. For example, a device with the ReadProperty service can read a value in another object; in the same manner, however, a value in one of its own objects can also be changed from the outside.

12.1.3.1.2.3.2 Prioritization

Prioritization for Output Objects

BACnet applications can access Objects through Services and change their properties. Access prioritization is required to regulate and organize access of various applications to the properties. The BACnet standard distinguishes between 16 different priority levels.

A priority can only refer to current values (Present_Value) of output objects. For this reason they are also called "command properties".

The processing sequences for individual applications can be changed using prioritization. Thus, the simultaneous access for several uses of the same objects is regulated (see following figure). The use on the highest priority level with the lowest number has priority.

Table 112: Priority Levels

Priority	Use
1 (high)	Manual safety control (Manual-Life Safety)
2	Automatic safety control (Automatic-Life Safety)
3	Freely available
4	Freely available
5	Use control for critical processes (Critical Equipment Control)
6	Time-limited switching on/off (Minimum On/Off)
7	Freely available
8	Manual settings (Manual Operator)
9	Freely available
10	Freely available
11	Freely available
12	Freely available
13	Freely available
14	Freely available
15	Freely available
16 (low)	Standard setting

Priority levels 3, 4, 7, and 9-16 may be freely allocated with use cases. In this way, certain controls can be assigned other priorities as needed (vacation planning, night operation, etc.).

Example:

If employees switch the lights on in their offices, a service executes this switching process with priority 8. For operation during vacations, it may be advisable to completely switch off certain technical devices and the lighting for the duration of the vacation. This action requires a higher priority. If a fire or another emergency occurs, the caretaker can control the lighting of the entire office complex and other measures with a general switch. Such emergency controls have the highest priority.

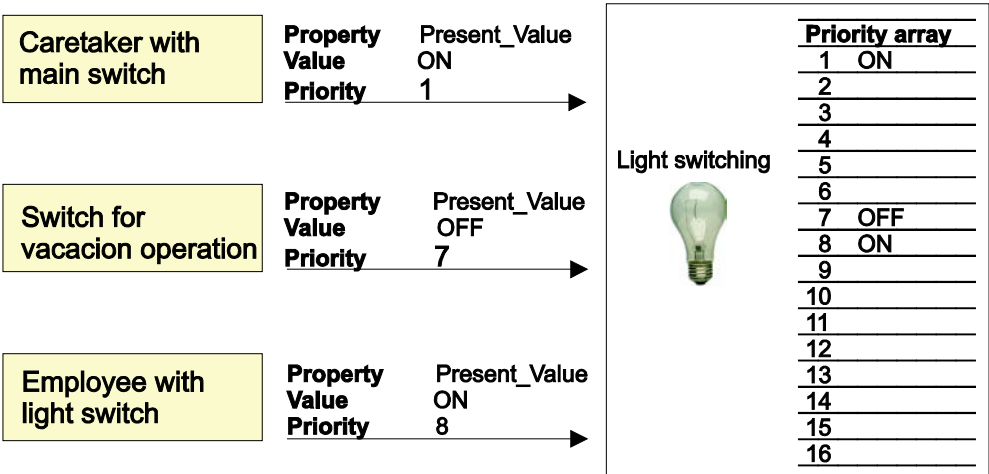


Figure 95: Examples of a Prioritization

Minimum On/Off

Some objects, such as the Binary Output Object, have the properties "Minimum_Off_Time" and "Minimum_On_Time". If these properties are assigned values (Unsigned32), these indicate within seconds how long the current state must be maintained before another can occur. In this way, the frequency of switching is controlled.

A minimum on or off time is, for example, used to control sun blinds for shading a building. We do not want the blinds to go up or down every minute depending on the cloud cover; rather, we want them to close during longer periods of sunshine and to be up in a secure position during bad weather or long-lasting storms.

Example:

Control of the sun blinds in an office complex is effected through the Write-Property services of three devices (switches). All three services deliver a priority parameter together with the command to the blinds to go up and down (UP/DOWN). This parameter is entered in the priority list in the object that controls the sun blinds. Now the value of the Present_Value of the object that features the highest priority (lowest number) is accepted. In this example, a 300 second Minimum_On_Time and a 600 second Minimum_Off_Time is assumed. This means that the minimum time that the sun blinds can stay in the highest position is 5 minutes, and only after this time has elapsed can new switching tasks be carried out. The minimum off time, in which the sun blinds are closed, must be at least 10 minutes before switching back.

Priority Array	
1	UP
2	
3	
4	
5	
6	UP
7	DOWN
8	
9	
10	
11	
12	
13	
14	
15	
16	

UP

The change to the value "UP", with priority "1", has precedence and is registered as the Present_Value in the object. The value "UP" is assigned priority level "6", which is intended for time-limited off and on switching. The blinds are pulled up. The time of the change in status is recorded and the 5 minutes of the minimum switching time begin.

Priority Array	
1	NULL
2	
3	
4	
5	
6	NULL
7	DOWN
8	
9	
10	
11	
12	
13	
14	
15	
16	

NULL

DOWN

If, after some time, the priority level "1" is overwritten with "NULL", the status of the minimum on/off switching time is also overwritten with NULL and reset. The value in the next higher priority level "7" is taken over as the "Present_Value" with "DOWN".

Priority Array	
1	
2	UP
3	
4	
5	
6	DOWN
7	DOWN
8	
9	
10	
11	
12	
13	
14	
15	
16	

UP

600 sec.

While the Minimum_Off_Time is running, no switching can take place. Not even a switching process with a higher priority of "2" can cause a change in status. Not until the Minimum_Off_Time of 10 minutes has passed will the value pick up the higher priority. In this case, the blinds go up ("UP").

To reset a task, a WriteProperty containing the value NULL must be resent. If all priority levels are assigned the value NULL, the standard value will be assumed.

12.1.3.1.2.4 Interoperability Area (IA)

The requirements for the overall operability of the system are divided into 5 areas. These areas establish functions for sub-areas of the requirements and serve as the basis for the evaluation of interoperability. They are called Interoperability Areas (IA):

Data Sharing - DS

For the system-wide, joint processing of data items, e.g. sensor information, target value and parameter changes, facility operation, etc., an agreement on data sharing is necessary.

Communication can be possible in both directions, both reading and writing.

Two types of communication are supported.

- Reading and writing through requests and tasks
- Event-oriented data transmission using COV/COS (Change of Value/State)

Example: DS-RP-A **Data Sharing ReadProperty - A**
The Client (A) makes the request to read a value (RP), to the Server (B).

Alarm and Event Notification - AE

Alarm and event processes are used to create and send notifications to certain recipients. Alarm confirmations are also managed and organized.

Whether an operation or alarm notification is to be generated can be optionally established through the object property "Notify_Type". This is necessary if intrinsic reporting is supported by the object.

An event can trigger special actions, represent a state that triggers an alarm, request a confirmation through an operator or just be registered/logged.

Two mechanisms for creating a notification are defined.

- Intrinsic Reporting
Relies on events internal to the object that are responsible for monitoring events and alarms.
- Rule-based notification (arithmetic change reporting)

Example: AE-N-A **Alarm and Event Notification - A**
The Client (A) processes alarms and notifications. The services ConfirmedEventNotification and UnconfirmedEventNotification are necessary for this. These services enable a report back to the initiator regarding the processability of the event notification.

Scheduling - SCHED

Actions that are to be managed with regard to time can be set through Scheduling. Two basic scheduling processes can be set.

- Weekly schedules
- Special schedules for exception days

Whether a schedule is used and which type is supported can be selected according to need.

Example: SCHED-A **Scheduling - A**
Device A changes schedule/calendar in Device B.
Device A must support DS-RP-A and DS-WP-A,
therefore read Device B and be allowed to write.

Trending - T

Time/value pairs are transmitted. Sampling takes place parameterized at narrow intervals so that exact data for immediate testing is delivered. The data are therefore not intended for long-term storage.

The recording of trends can take place through the COV service "Transmit with change in value" or by periodic query.

Notifications are issued through intrinsic or rule-based reporting.

Example: T-VMT-A **Trending Viewing and Modifying Trends - A**
Device A displays the trending records of Device B
and alters the Trend-Log Parameter there.

Device Management - DM

Information on state, presence and availability of BACnet devices is exchanged. Communication with certain devices can also be enabled and disabled. Through DM, clock time in the entire system can be synchronized. Software can also be restarted on demand. There are also diverse diagnostic and access possibilities.

Example: DM-DDB-A **Device-Management-Dynamic-Device-Binding-A**
Device A obtains information on other devices within
the network and interprets device information

Each IA is composed of several BIBBs. These contain functions that vary depending on device profile (see section "Device Profile").

12.1.3.1.2.5 BACnet Interoperability Building Blocks - BIBBs

A BIBB defines which elements of the BACnet protocol must be implemented in the device. Compared to the BIBBs from different manufacturers, common features represent a basis for interoperability between the devices. BIBBs form function blocks for the respective Interoperability Areas (IA).

Example:

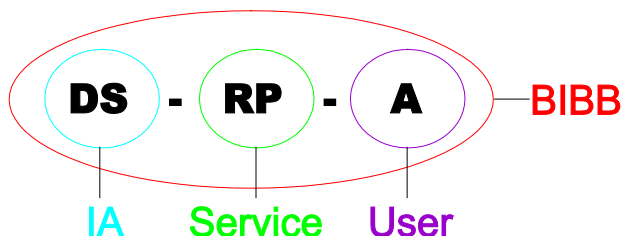


Figure 96: Example BACnet Interoperability Building Block

Meaning:

- **DS (DataSharing)** is the IA of the BIBB
- **RP (ReadProperty)** designates the executable service
- **A (Data Requester)** identifies the Client which, in this case, makes a read request to a Server (B for data holder).

12.1.3.1.2.6 Protocol Implementation Conformance Statement - PICS

The Protocol Implementation Conformance Statement (PICS) describes objects, capabilities and functions that the BACnet device supports. This is a standard document that must be filled in by the manufacturer. When linking different systems, the PICS and the BIBBs and other device features contained therein are compared with one another. At a minimum, a BACnet PICS must convey the following information:

- Basic information identifying the vendor and describing the BACnet device
- BACnet Interoperability Building Blocks - BIBBS - supported by the device
- The standardized BACnet device profile to which the device conforms (e.g. the device profile of the BACnet Building Controller - B-BC)
- All non-standard application services that are supported along with an indication for each service of whether the device can initiate the service request, respond to a service request, or both. All non-standard application services that are supported along with an indication for each service of whether the device can initiate the service request, respond to a service request, or both
- A list of all standard and proprietary objects that are supported
- For each object type supported,
 - any optional properties that are supported,
 - which properties can be written-to using BACnet services,
 - if the objects can be dynamically created or deleted using BACnet services,
 - any restrictions on the range of data values for properties.
- Options, both real and virtual, for the supported data link layer
- Whether segmented queries are supported
- Whether segmented responses are supported

Information



Additional Information

The PICS document for the BACnet/IP Controller 750-831 is provided at the website <http://www.wago.com> under Documentation → WAGO-I/O-SYSTEM 750 → Fieldbus couplers and programmable fieldbus controllers → 750-831 → Additional information

12.1.3.1.2.7 Device Profile

There are eight different device profiles defined in Annex L of the BACnet Standard. They are divided up according to the minimum BIBBs to be supported in each IA and enable a rapid and authoritative comparison between different BACnet devices.

- **BACnet Operator Workstation (B-OWS)**
Operator interface/building control technology used for the management and configuration tasks of the BACnet system, but does not take over any direct control and regulation tasks.
- **BACnet Operator Display (B-OD)**
Comparable to a B-OWS, but with basic functionality only, e.g. for LCD screens and mobile devices.
- **BACnet Application Specific Controller (B-AWS)**
As further development of a B-OWS, a B-AWS handles control and regulating tasks in addition to performance measurements.
- **BACnet Building Controller (B-BC)**
Freely programmable automation system that takes over a multitude of automation/regulation and control tasks.
- **BACnet Advanced Application Controller (B-AAC)**
Corresponds to a configurable application controller that takes over specially prefabricated tasks, but with less available means than the B-BC.
- **BACnet Application Specific Controller (B-ASC)**
Corresponds to an automation device/controller, but with fewer available means than the B-AAC. Functions as a control and regulation unit for special applications and programs and their parameterization.
- **BACnet Smart Actuator (B-SA)**
Intelligent, network-capable field device for switching and actuating
- **BACnet Smart Sensor (B-SS)**
Intelligent, network-capable field device as sensor

For each of the listed device profiles, there is a certain set of BIBBs to implement as a minimum requirement in the BACnet/IP controller. This minimum requirement represents the capabilities of the device in the form of function blocks (see following table) and makes devices comparable with each other.

Table 113: IA/BIBB of the Standardized Device Profiles

B-AWS	B-OWS	B-OD	B-BC	B-AAC	B-ASC	B-SA	B-SS
Data Sharing							
DS-RP-A,B DS-RPM-A DS-WP-A DS-WPM-A DS-AV-A DS-AM-A	DS-RP-A,B DS-RPM-A DS-WP-A DS-WPM-A DS-V-A DS-M-A	DS-RP-A,B DS-WP-A DS-V-A DS-M-A	DS-RP-A,B DS-RPM-A,B DS-WP-A,B DS-WPM-B	DS-RP-B DS-RPM-B DS-WP-B DS-WPM-B	DS-RP-B DS-WP-B	DS-RP-B DS-WP-B	DS-RP-B
Alarm & Event Management							
AE-N-A AE-ACK-A AE-AS-A AE-AVM-A AE-AVN-A AE-ELVM-A ²	AE-N-A AE-ACK-A AE-AS-A AE-VM-A AE-VN-A	 AE-VN-A	AE-N-I-B AE-ACK-B AE-INFO-B AE-ESUM-B	AE-N-I-B AE-ACK-B AE-INFO-B			
Scheduling							
SCHED-AVM-A	SCHED-VM-A		SCHED-E-B	SCHED-I-B			
Trending							
T-AVM-A	T-V-A		T-VMT-I-B T-ATR-B				
Device & Network Management							
DM-DDB-A,B DM-ANM-A DM-ADM-A DM-DOB-B DM-DCC-A DM-MTS-A DM-OCD-A DM-RD-A DM-BR-A	DM-DDB-A,B DM-DOB-B DM-MTS-A	DM-DDB-A,B DM-DOB-B	DM-DDB-A,B DM-DOB-B DM-DCC-B DM-TS-B or DM-UTC-B DM-RD-B DM-BR-B	DM-DDB-B DM-DOB-B DM-DCC-B DM-TS-B or DM-UTC-B DM-RD-B	DM-DDB-B DM-DOB-B DM-DCC-B DM-DOB-B ¹	DM-DDB-B ¹ DM-DOB-B ¹	DM-DDB-B ¹ DM-DOB-B ¹

¹ Not required if the device is a BACnet MS/TP slave² Not required for devices claiming conformance to a Protocol_Revision less than 7.

12.1.3.1.2.8 Data Types

Services utilize communication units, so-called "Application Layer Protocol Data Units" (APDUs) that are defined in an abstract ASN.8824 syntax in accordance with ISO Standard 1 to ensure uniform data transfer. Both simple data types like BOOLEAN, INTEGER and REAL as well as special BACnet data types are used. These can in turn be nested as SEQUENCE, SET, CHOICE, etc.



Information

Additional Information

You can find the device-specific data representation of the BACnet Objects and Services in the documentation of the BACnet Library "BACnet_xx.lib" at the website <http://www.wago.com> under Service Downloads → Building Automation → BACnet Downloads → Software

12.1.3.1.2.9 BACnet Components in Overview

The following illustration explains the connection between the previously named components using the example of a BACnet Building Controller (B-BC).

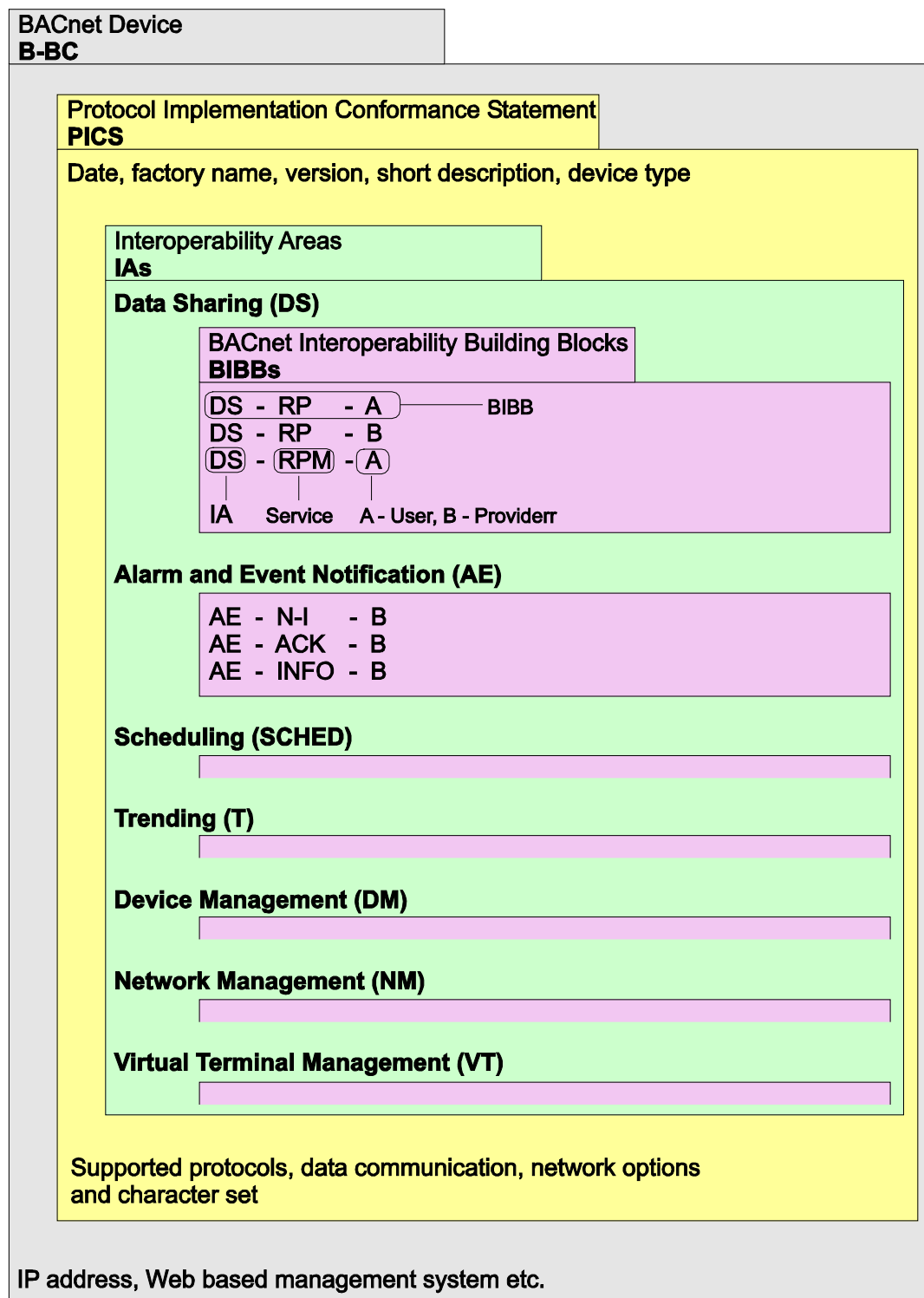


Figure 97: Connection between BACnet Components

12.1.3.1.3 Scope of Use in Building Automation

The use of BACnet devices has the advantage that many different devices and networks can be simply connected with each other and administrated. In doing so, the BACnet Protocol covers communication on the automation and management levels.

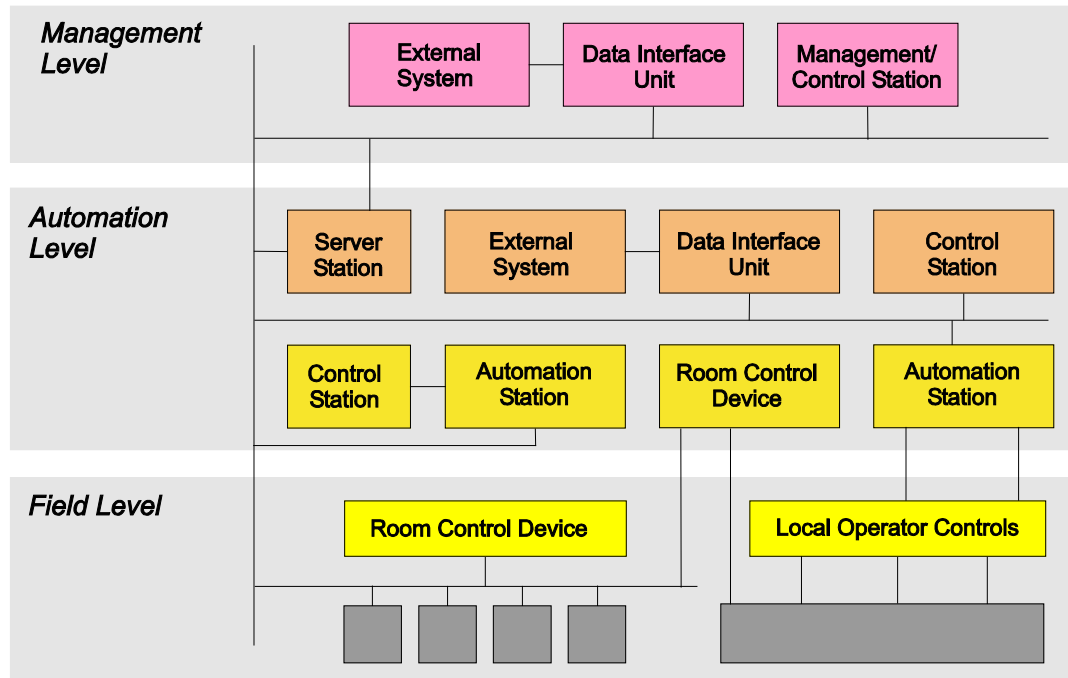


Figure 98: Use of BACnet in Building Automation

12.1.3.1.3.1 Management Level

On this level, the processes of building automation are continuously analyzed, improved, stored and evaluated. Functions of this level are also necessary for transmitting input and output values to the management levels of foreign systems. For communication between heterogeneous systems, these must be linked in a manner that is interoperable, i.e. correspond in the communication structure (see sections "Interoperability Area" and "BACnet Interoperability Building Blocks").

Application notes

- Facility Management
- Finance and Personnel Planning
- Energy Management
- Servicing

12.1.3.1.3.2 Automation Level

The automation level includes alarm organization and the operation of systems.

Areas of use

- Monitoring
- Optimizing
- Control
- Regulating
- Reporting

12.1.3.1.3.3 Field Level

On this level, individual devices and alarm notifications, e.g. the recording of states and measured values, are controlled.

Application notes

- Trade shows
- Switching
- Setting

12.1.3.1.4 BACnet in the ISO/OSI Model

BACnet works in an object-oriented manner on the basis of the ISO/OSI Model. Data communication takes place on three levels.

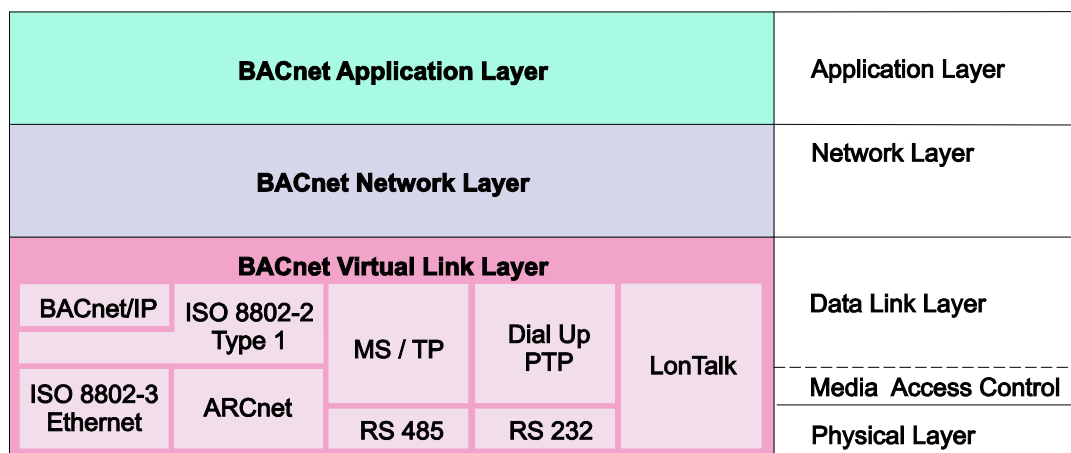


Figure 99: BACnet Layers in the ISO/OSI Model

BACnet Application Layer

This layer represents the application layer and interface with the outside. Here, communication takes place via BACnet.

BACnet Network Layer

On this relay level, data transport via BACnet, PROFIBUS FMS, World-FIP and/or EIBnet takes place.

BACnet Virtual Link Layer (BVLL)

This layer includes the backup and bit transmission of data. The data can be transmitted using protocols and connection possibilities that are established in the Standard. The following technologies are supported:

- **ARCNET (Attached Resources Computer Network)** as a networking technology for local networks (mostly USA)
- **ETHERNET** as a transmission technology in local networks; specifies transmission media of the physical layer (layer 1 of the OSI Model) and also serves as transmission backup in layer 2 of the OSI Model
- **BACnet/IP** uses the UDP Protocol (User Datagram Protocol) for data transmission
- **PTP (Point-To-Point)** via RS-232 as serial connections; enables point-to-point connections between two participants, e.g. over a telephone line.
- **MS/TP (Master-Slave/Token-Passing) via RS-485** as a serial network for long lengths of line and simple construction and wiring
- **LonTalk ANSI/EIA709.1** as a fieldbus from the company Echelon with decentralized control; BACnet uses the transport layers of LonTalk (e.g. FTT-10 over 2-wire)

12.1.3.1.5 BACnet in the Network

There are 2 possibilities for sending reports over networks that are based on the Internet protocol (IP)

- IP Message Tunneling
- BACnet/IP

12.1.3.1.5.1 IP Message Tunneling

Devices that do not communicate over BACnet/IP or that do not use the interfaces specified by the BACnet Standard for communication need a BACnet Tunneling Router (BTRs). Since the functioning is described in Annex H of the Standard, these routers are also called "Annex H routers".

IP tunneling between the different communication technologies takes place over routing tables with a combination of BACnet network numbers and IP addresses.

Sending a report from one device A to a device B requires the BACnet Protocol Annex H router in both local networks (see following figure).

The Annex H router for network 1 transfers the BACnet message to a UDP (User Datagram Protocol) frame and transmits the message over standard IP links, or over the Internet to the Annex H router in network 2.

This unpacks the incoming data packet and sends the report over the BACnet Protocol to the target device B.

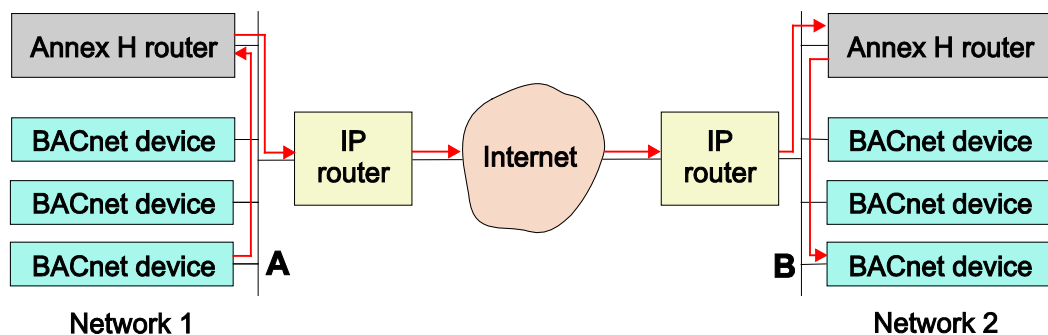


Figure 100: Communication over an Annex-H router

The advantage of this type of communication via BTRs is the economical delivery costs. Also, the BACnet devices do not have to be IP-capable. BTRs are frequently used in existing BACnet networks that have a link to IP networks, to an intranet or to the Internet.

A disadvantage of this method is the high data traffic on the line, for each report is sent twice over the network - once as a BACnet and once as an IP report.

12.1.3.1.5.2 BACnet/IP

For data transmission via BACnet/IP, each individual BACnet device in a sub-net must be IP-capable, i.e. has its own IP address and an IP Protocol Stack. In this way, devices can communicate directly with each other. No tunneling routers are necessary (see following figure).

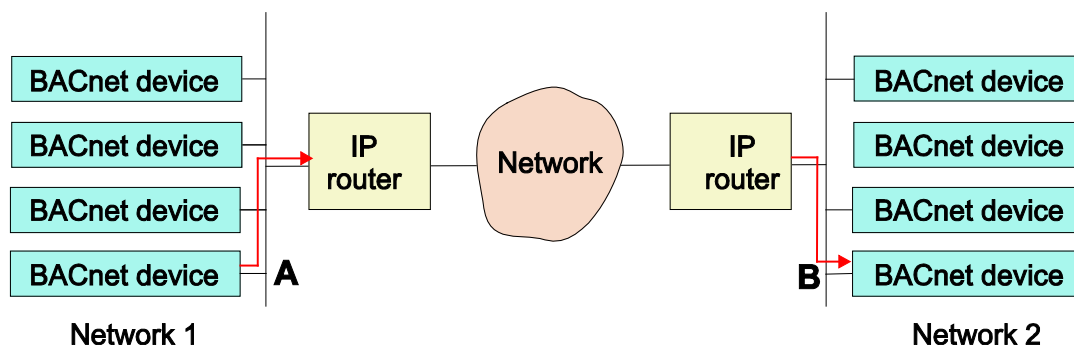


Figure 101: Communication via BACnet/IP

To control function on the lowest level, the BACnet Virtual Link Layer (BVLL) was developed. This provides a report image that is suitable for the specific features of IP networks. The advantage of the BVLL concept is the simple adjustment of the BVLL control information, so that nearly every network technology can be converted into the specific BACnet structure.

12.1.3.1.5.3 BACnet/IP and Unicast/Broadcast

To send a report from one device to another (Unicast), BACnet/IP devices do not need any tunneling routers and can communicate directly over the Internet with each other.

Broadcast reports, on the other hand, are usually blocked by IP routers. For some BACnet functions, this form of "report transmission to all" is necessary, however, e.g. for a "Who Is" request. So, either interposed routers have a broadcast mode and forward the report, or special routers are interposed that support this communication - "BACnet Broadcast Management Devices" (BBMD).

A BBMD works in a manner similar to that of an Annex H router, but only takes over the sending of broadcast IP reports by converting broadcast IP reports to unicast reports and sending them over the Internet (see following figure).

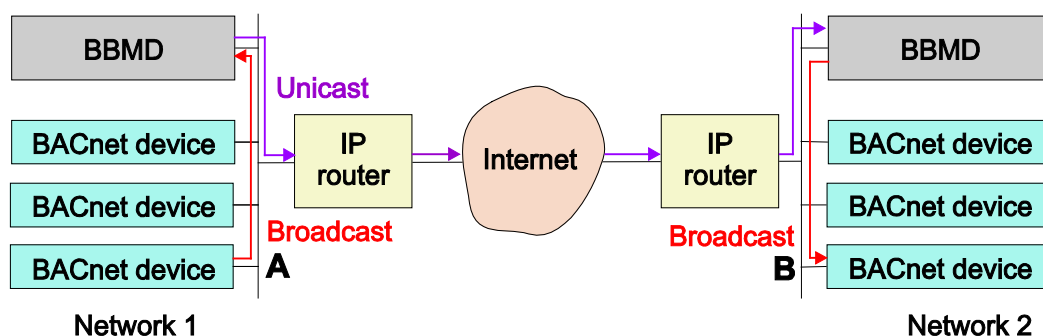


Figure 102: Sending a Broadcast Report

12.1.3.1.5.4 BACnet/IP in Foreign Networks

If a device is connected to one subnet and would like to receive broadcasts from another subnet or send to another subnet, BBMDs are required in both subnets.

For a device to communicate that does not have a BBMD in its own (sub)network and therefore becomes a Foreign Device (FD), the other BACnet/IP (sub)networks must have BBMD/FDs (see following figure). The foreign devices register with this BBMD/FD using their IP addresses. In this way, broadcasts from other subnets can be received or can be sent to these net-works.

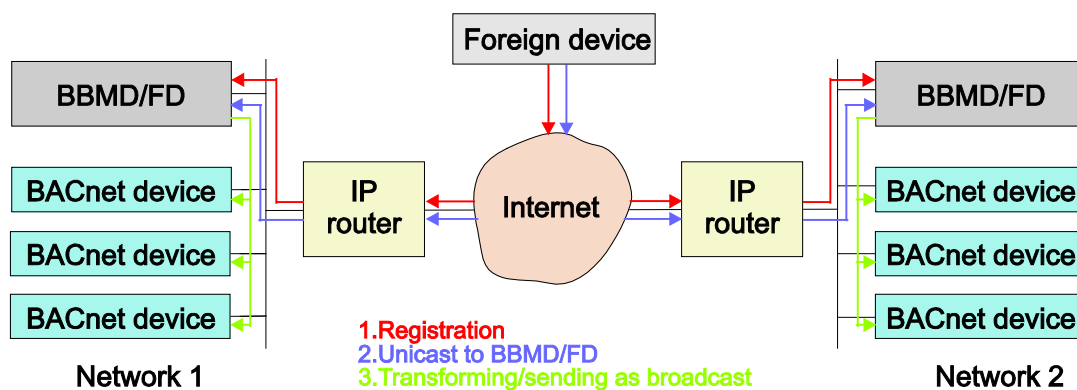


Figure 103: Registration of a Foreign Device

12.2 MODBUS Functions

12.2.1 General

MODBUS is a manufacturer-independent, open fieldbus standard for diverse applications in manufacturing and process automation.

The MODBUS protocol is implemented according to the current Internet Draft of the IETF (Internet Engineering Task Force) and performs the following functions:

- Transmission of the process image
- Transmission of the fieldbus variables
- Transmission of different settings and information on the coupler/controller

The data transmission in the fieldside takes place via TCP and via UDP.

The MODBUS/TCP protocol is a variation of the MODBUS protocol, which was optimized for communication via TCP/IP connections.

This protocol was designed for data exchange in the field level (i.e. for the exchange of I/O data in the process image).

All data packets are sent via a TCP connection with the port number 502.

MODBUS/TCP segment

The general MODBUS/TCP header is as follows:

Table 114: MODBUS/TCP Header

Byte	0	1	2	3	4	5	6	7	8 ... n
	Identifier (entered by receiver)		Protocol- identifier (is always 0)		Length field (High byte, low byte)		Unit identifier (Slave address)	MODBUS function code	Data



Information

Additional Information

The structure of a datagram is specific for the individual function. Refer to the descriptions of the MODBUS Function codes.

For the MODBUS protocol 30 connections are made available over TCP. Thus it allows digital and analog output data to be directly read out at a fieldbus node and special functions to be executed by way of simple MODBUS function codes from 30 stations simultaneously.

For this purpose a set of MODBUS functions from the Open MODBUS/TCP specification is realized.



Information

More information

More information on the “Open MODBUS/TCP specification” you can find in the Internet: www.modbus.org.

Therefore the MODBUS protocol based essentially on the following basic data types:

Table 115: Basic Data Types of MODBUS Protocol

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

For each basic data type one or more function codes are defined.

These functions allow digital or analog input and output data, and internal variables to be set or directly read out of the fieldbus node.

Table 116: List of the MODBUS Functions in the Fieldbus Controller

Function code	Function	Access method and description	Access to resources
FC1 0x01	Read Coils	Reading of several output bits	R: Process image, PFC variables
FC2 0x02	Read Discrete Inputs	Reading of several input bits	R: Process image, PFC variables
FC3 0x03	Read Holding Registers	Reading of several input registers	R: Process image, PFC variables, internal variables, NOVRAM
FC4 0x04	Read Input Registers	Reading of several input registers	R: Process image, PFC variables, internal variables, NOVRAM
FC5 0x05	Write Single Coil	Writing of an individual output bit	W: Process image, PFC variables
FC6 0x06	Write Single Register	Writing of an individual output register	W: Process image, PFC variables, internal variables, NOVRAM
FC11 0x0B	Get Comm Event Counters	Communication event counter	R: None
FC15 0x0F	Write Multiple Coils	Writing of several output bits	W: Process image, PFC variables
FC16 0x10	Write Multiple Registers	Writing of several output registers	W: Process image, PFC variables, internal variables, NOVRAM
FC22 0x16	Mask Write Register	Writing of several bits of an individual output register by mask	W: Process image, PFC variables, NOVRAM

FC23 0x17	Read/Write Multiple Registers	Reading and writing of several output registers	R/W: Process image, PFC variables, NOVRAM

To execute a desired function, specify the respective function code and the address of the selected input or output data.

Note



Note the number system when addressing!

The examples listed use the hexadecimal system (i.e.: 0x000) as their numerical format. Addressing begins with 0. The format and beginning of the addressing may vary according to the software and the control system. All addresses then need to be converted accordingly.

12.2.2 Use of the MODBUS Functions

The example below uses a graphical view of a fieldbus node to show which MODBUS functions can be used to access data of the process image.

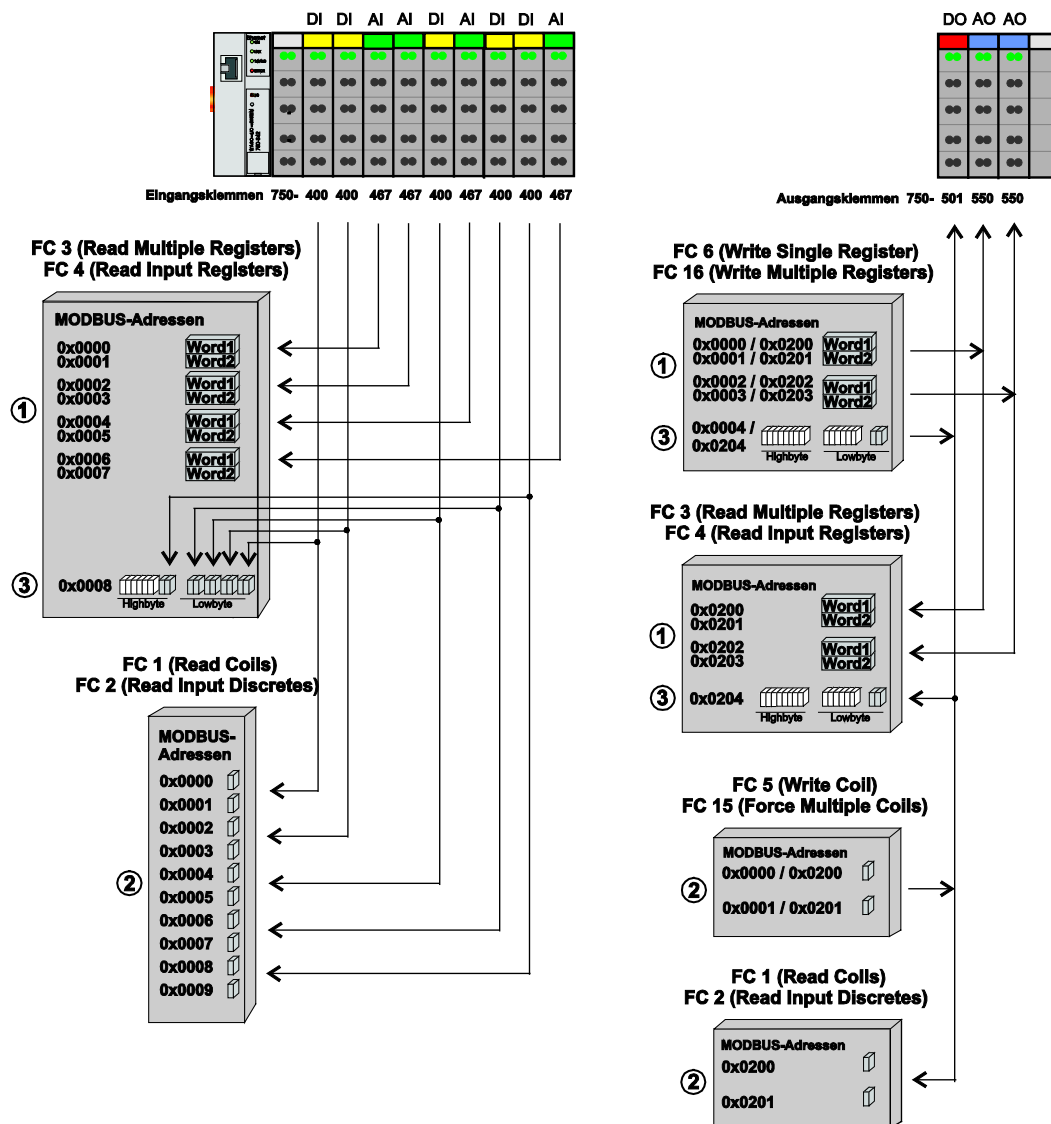


Figure 104: Use of the MODBUS Functions



Note

Use register functions to access analog signals and coil functions to access binary signals!

It is recommended that analog data be accessed with register functions ① and digital data with coil functions ②. If reading or writing access to binary signals is performed via register functions ③, an address shift may occur as soon as further analog modules are operated on the coupler/controller.

12.2.3 Description of the MODBUS Functions

All MODBUS functions are executed as follows:

1. A MODBUS TCP master (e.g., a PC) makes a request to the WAGO fieldbus node using a specific function code based on the desired operation..
2. The WAGO fieldbus node receives the datagram and then responds to the master with the proper data, which is based on the master's request.

If the WAGO fieldbus node receives an incorrect request, it sends an error datagram (Exception) to the master.

The exception code contained in the exception has the following meaning:

Table 117: Exception Codes

Exception code	Meaning
0x01	Illegal function
0x02	Illegal data address
0x03	Illegal data value
0x04	Slave device failure

The following chapters describe the datagram architecture of request, response and exception with examples for each function code.



Note

Reading and writing of outputs via FC1 to FC4 is also possible by adding an offset!

In the case of the read functions (FC1 ... FC4) the outputs can be additionally written and read back by adding an offset of 200hex (0x0200) to the MODBUS addresses in the range of [0_{hex} ... FF_{hex}] and an offset of 1000_{hex} (0x01000) to the MODBUS addresses in the range of [6000_{hex} ... 62FC_{hex}].

12.2.3.1 Function Code FC1 (Read Coils)

This function reads the status of the input and output bits (coils) in a slave device.

Request

The request specifies the reference number (starting address) and the bit count to read.

Example: Read output bits 0 to 7.

Table 118: Request of Function Code FC1

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x01
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0008

Response

The current values of the response bits are packed in the data field. A binary 1 corresponds to the ON status and a 0 to the OFF status. The lowest value bit of the first data byte contains the first bit of the request. The others follow in ascending order. If the number of inputs is not a multiple of 8, the remaining bits of the last data byte are filled with zeroes (truncated).

Table 119: Response of Function Code FC1

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x01
Byte 8	Byte count	0x01
Byte 9	Bit values	0x12

The status of the inputs 7 to 0 is shown as byte value 0x12 or binary 0001 0010. Input 7 is the bit having the highest significance of this byte and input 0 the lowest value.

The assignment is thus made from 7 to 0 as follows:

Table 120: Assignment of Inputs

	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF
Bit	0	0	0	1	0	0	1	0
Coil	7	6	5	4	3	2	1	0

Exception

Table 121: Exception of Function Code FC1

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x81
Byte 8	Exception code	0x01 or 0x02

12.2.3.2 Function Code FC2 (Read Discrete Inputs)

This function reads the input bits from a slave device.

Request

The request specifies the reference number (starting address) and the bit count to be read.

Example: Read input bits 0 to 7

Table 122: Request of Function Code FC2

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x02
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0008

Response

The current value of the requested bits are packed into the data field. A binary 1 corresponds to the ON status and a 0 the OFF status. The lowest value bit of the first data byte contains the first bit of the inquiry. The others follow in an ascending order. If the number of inputs is not a multiple of 8, the remaining bits of the last data byte are filled with zeroes (truncated).

Table 123: Response of Function Code FC2

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x02
Byte 8	Byte count	0x01
Byte 9	Bit values	0x12

The status of the inputs 7 to 0 is shown as a byte value 0x12 or binary 0001 0010. Input 7 is the bit having the highest significance of this byte and input 0 the lowest value. The assignment is thus made from 7 to 0 as follows:

Table 124: Assignment of Inputs

	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF
Bit	0	0	0	1	0	0	1	0
Coil	7	6	5	4	3	2	1	0

Exception

Table 125: Exception of Function Code FC2

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x82
Byte 8	Exception code	0x01 or 0x02

12.2.3.3 Function Code FC3 (Read Multiple Registers)

This function reads the contents of holding registers from a slave device in word format.

Request

The request specifies the reference number (start register) and the word count (register quantity) of the registers to be read. The reference number of the request is zero based, therefore, the first register starts at address 0.

Example: Read registers 0 and 1.

Table 126: Request of Function Code FC3

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x03
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002

Response

The reply register data is packed as 2 bytes per register. The first byte contains the higher value bits, the second the lower values.

Table 127: Response of Function Code FC3

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x03
Byte 8	Byte count	0x04
Byte 9, 10	Value register 0	0x1234
Byte 11, 12	Value register 1	0x2345

The contents of register 0 are displayed by the value 0x1234 and the contents of register 1 is 0x2345.

Exception

Table 128: Exception of Function Code FC3

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x83
Byte 8	Exception code	0x01 or 0x02

12.2.3.4 Function Code FC4 (Read Input Registers)

This function reads contents of input registers from the slave device in word format.

Request

The request specifies a reference number (start register) and the word count (register quantity) of the registers to be read. The reference number of the request is zero based, therefore, the first register starts at address 0.

Example: Read registers 0 and 1

Table 129: Request of Function Code FC4

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x04
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002

Response

The register data of the response is packed as 2 bytes per register. The first byte has the higher value bits, the second the lower values.

Table 130: Response of Function Code FC4

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x04
Byte 8	Byte count	0x04
Byte 9, 10	Value register 0	0x1234
Byte 11, 12	Value register 1	0x2345

The contents of register 0 are shown by the value 0x1234 and the contents of register 1 is 0x2345.

Exception

Table 131: Exception of Function Code FC4

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x84
Byte 8	Exception code	0x01 or 0x02

12.2.3.5 Function Code FC5 (Write Coil)

This function writes a single output bit to the slave device.

Request

The request specifies the reference number (output address) of output bit to be written. The reference number of the request is zero based; therefore, the first coil starts at address 0.

Example: Turn ON the second output bit (address 1)

Table 132: Request of Function Code FC5

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x05
Byte 8, 9	Reference number	0x0001
Byte 10	ON/OFF	0xFF
Byte 11		0x00

Response

Table 133: Response of Function Code FC5

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x05
Byte 8, 9	Reference number	0x0001
Byte 10	Value	0xFF
Byte 11		0x00

Exception

Table 134: Exception of Function Code FC5

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01, 0x02 or 0x03

12.2.3.6 Function Code FC6 (Write Single Register)

This function writes the value of one single output register to a slave device in word format.

Request

The request specifies the reference number (register address) of the first output word to be written. The value to be written is specified in the "Register Value" field. The reference number of the request is zero based; therefore, the first register starts at address 0.

Example: Write a value of 0x1234 to the second output register

Table 135: Request of Function Code FC6

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x06
Byte 8, 9	Reference number	0x0001
Byte 10, 11	Register value	0x1234

Response

The reply is an echo of the inquiry.

Table 136: Response of Function Code FC6

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x06
Byte 8, 9	Reference number	0x0001
Byte 10, 11	Register value	0x1234

Exception

Table 137: Exception of Function Code FC6

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.7 Function Code FC11 (Get Comm Event Counter)

This function returns a status word and an event counter from the slave device's communication event counter. By reading the current count before and after a series of messages, a master can determine whether the messages were handled normally by the slave.

Following each successful new processing, the counter counts up. This counting process is not performed in the case of exception replies, poll commands or counter inquiries.

Request

Table 138: Request of Function code FC11

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0002
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x0B

Response

The reply contains a 2-byte status word and a 2-byte event counter. The status word only contains zeroes.

Table 139: Response of Function Code FC11

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x0B
Byte 8, 9	Status	0x0000
Byte 10, 11	Event count	0x0003

The event counter shows that 3 (0x0003) events were counted.

Exception

Table 140: Exception of Function Code FC 11

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.8 Function Code FC15 (Write Multiple Coils)

This function sets a sequence of output bits to 1 or 0 in a slave device. The maximum number is 256 bits.

Request

The request message specifies the reference number (first coil in the sequence), the bit count (number of bits to be written), and the output data. The output coils are zero-based; therefore, the first output point is 0.

In this example 16 bits are set, starting with the address 0. The request contains 2 bytes with the value 0xA5F0, or 1010 0101 1111 0000 in binary format.

The first data byte transmits the value of 0xA5 to the addresses 7 to 0, whereby 0 is the lowest value bit. The next byte transmits 0xF0 to the addresses 15 to 8, whereby the lowest value bit is 8.

Table 141: Request of Function Code FC15

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0009
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x0F
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0010
Byte 12	Byte count	0x02
Byte 13	Data byte1	0xA5
Byte 14	Data byte2	0xF0

Response

Table 142: Response of Function Code FC15

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x0F
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0010

Exception

Table 143: Exception of Function Code FC15

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x8F
Byte 8	Exception code	0x01 or 0x02

12.2.3.9 Function Code FC16 (Write Multiple Registers)

This function writes a sequence of registers in a slave device in word format.

Request

The Request specifies the reference number (starting register), the word count (number of registers to write), and the register data. The data is sent as 2 bytes per register. The registers are zero-based; therefore, the first output is at address 0. Example: Set data in registers 0 and 1

Table 144: Request of Function Code FC16

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x000B
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002
Byte 12	Byte count	0x04
Byte 13, 14	Register value 1	0x1234
Byte 15, 16	Register value 2	0x2345

Response

Table 145: Response of Function Code FC16

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002

Exception

Table 146: Exception of Function Code FC16

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.10 Function Code FC22 (Mask Write Register)

This function manipulates individual bits within a register using a combination of an AND mask, an OR mask, and the register's current content.

Request

Table 147: Request of Function Code FC22

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0002
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x16
Byte 8, 9	Reference number	0x0000
Byte 10, 11	AND mask	0x0000
Byte 12, 13	OR mask	0xAAAA

Response

Table 148: Response of Function Code FC22

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	AND mask	0x0000
Byte 12, 13	OR mask	0xAAAA

Exception

Table 149: Exception of Function Code FC22

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.11 Function Code FC23 (Read/Write Multiple Registers)

This function performs a combination of a read and write operation in a single request. The function can write the new data to a group registers, and then return the data of a different group. The write operation is performed before the read.

Request

The reference numbers (addresses) are zero-based in the request message; therefore, the first register is at address 0.

The request message specifies the registers to read and write. The data is sent as 2 bytes per register.

Example: The data in register 3 is set to value 0x0123, and values 0x0004 and 0x5678 are read out of the two registers 0 and 1.

Table 150: Request of Function Code FC23

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x000D
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x17
Byte 8, 9	Reference number for read	0x0000
Byte 10, 11	Word count for read (1...125)	0x0002
Byte 12, 13	Reference number for write	0x0003
Byte 14, 15	Word count for write (1...100)	0x0001
Byte 16	Byte count (2 x word count for write)	0x02
Byte 17...(B+16)	Register values (B = Byte count)	0x0123

Response

Table 151: Response of Function Code FC23

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x17
Byte 8	Byte count (2 x word count for read)	0x04
Byte 9...(B+1)	Register values (B = Byte count)	0x0004 or 0x5678

Exception

Table 152: Exception of Function Code FC23

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x97
Byte 8	Exception code	0x01 or 0x02



Note

Note that if the register ranges overlap, the results are undefined!
If register areas for read and write overlap, the results are undefined.

12.2.4 MODBUS Register Mapping

The following tables display the MODBUS addressing and the corresponding IEC61131 addressing for the process image, the PFC variables, the NOVDRAM data, and the internal variables is represented.

Via the register services the states of the complex and digital I/O modules can be determined or changed.

Register Access Reading (with FC3, FC4 and FC23)

Table 153: Register Access Reading (with FC3, FC4 and FC23)

MODBUS address		IEC 61131	Memory range
[dec]	[hex]	address	
0...255	0x0000...0x00FF	%IW0...%IW255	Physical input area (1) First 256 words of physical input data
256...511	0x0100...0x01FF	%QW256...%QW511	PFC OUT area Volatile PFC output variables
512...767	0x0200...0x02FF	%QW0...%QW255	Physical output area (1) First 256 words of physical output data
768...1023	0x0300...0x03FF	%IW256...%IW511	PFC IN area Volatile PFC input variables
1024...4095	0x0400...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...12287	0x1000...0x2FFF	-	Configuration register (see following chapter "Configuration Functions") Not with FC23!
12288...24575	0x3000...0x5FFF	%MW0...%MW12287	24 kB retain memory *) *) In target system settings RETAIN to 0 kB, bit memory to 24 kB
24576...25340	0x6000...0x62FC	%IW512...%IW1275	Physical input area (2) Additional 764 words physical input data
25341...28671	0x62FD...0x6FFF	-	MODBUS exception: "Illegal data address"
28672...29435	0x7000...0x72FB	%QW512...%QW1275	Physical output area (2) Additional 764 words physical output data
29436...32767	0x72FC...0x7FFF	-	MODBUS exception: "Illegal data address"
32768...34815	0x8000...0x87FF	-	NOVDRAM
34816...36863	0x8800...0x8FFF	-	BACnet-specific use, no MODBUS access
36864...65535	0x9000...0xFFFF	-	MODBUS exception: "Illegal data address"

Register Access Writing (with FC6, FC16, FC22 and FC23)

Table 154: Register Access Writing (with FC6, FC16, FC22 and FC23)

MODBUS address		IEC 61131	Memory range
[dec]	[hex]	address	
0...255	0x0000...0x00FF	%QW0...%QW255	Physical output area (1) First 256 words of physical output data
256...511	0x0100...0x01FF	%IW256...%IW511	PFC IN area Volatile PFC input variables
512...767	0x0200...0x02FF	%QW0...%QW255	Physical output area (1) First 256 words of physical output data
768...1023	0x0300...0x03FF	%IW256...%IW511	PFC IN area Volatile PFC input variables
1024...4095	0x0400...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...12287	0x1000...0x2FFF	-	Configuration register (see following chapter „Configuration Functions“) Not with FC22 and FC23!
12288...24575	0x3000...0x5FFF	%MW0...%MW12287	24 kB retain memory *) *) In target system settings RETAIN to 0 kB, bit memory to 24 kB
24576...25340	0x6000...0x62FC	%QW512...%QW1275	Physical output area (2) Additional 764 words physical output data
25341...28671	0x62FD...0x6FFF	-	MODBUS exception: "Illegal data address"
28672...29435	0x7000...0x72FB	%QW512...%QW1275	Physical output area (2) Additional 764 words physical output data
29436...32767	0x72FC...0x7FFF	-	MODBUS exception: "Illegal data address"
32768...34815	0x8000...0x87FF	-	NOVRAM
34816...36863	0x8800...0x8FFF	-	BACnet-specific use, no MODBUS access
36864...65535	0x9000...0xFFFF	-	MODBUS exception: "Illegal data address"

The digital MODBUS services (coil services) are bit accesses, with which only the states of digital I/O modules can be determined or changed. Complex I/O modules are not attainable with these services and so they are ignored. Because of this the addressing of the digital channels begins again with 0, so that the MODBUS address is always identical to the channel number, (i.e. the digital input no. 47 has the MODBUS address "46").

Bit Access Reading (with FC1 and FC2)

Table 155: Bit Access Reading (with FC1 and FC2)

MODBUS address		Memory range	Description
[dec]	[hex]		
0...511	0x0000...0x01FF	Physical input area (1)	First 512 digital inputs
512...1023	0x0200...0x03FF	Physical output area (1)	First 512 digital outputs
1024...4095	0x0400...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...8191	0x1000...0x1FFF	%QX256.0...%QX511.15	PFC OUT area Volatile PFC output variables
8192...12287	0x2000...0x2FFF	%IX256.0...%IX511.15	PFC IN area Volatile PFC input variables
12288...32767	0x3000...0x7FFF	%MX0...%MX1279.15	NOVRAM 2 kB retain memory (max. 24 kB)
32768...34295	0x8000...0x85F7	Physical input area (2)	Starts with the 513 th and ends with the 2039 th digital input
34296...36863	0x85F8...0x8FFF	-	MODBUS exception: "Illegal data address"
36864...38391	0x9000...0x95F7	Physical output area (2)	Starts with the 513 th and ends with the 2039 th digital output
38392...65535	0x95F8...0xFFFF	-	MODBUS exception: "Illegal data address"

Bit Access Writing (with FC5 and FC15)

Table 156: Bit Access Writing (with FC5 and FC15)

MODBUS address		Memory range	Description
[dec]	[hex]		
0...511	0x0000...0x01FF	Physical output area (1)	First 512 digital outputs
512...1023	0x0200...0x03FF	Physical output area (1)	First 512 digital outputs
1024...4095	0x0400...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...8191	0x1000...0x1FFF	%IX256.0...%IX511.15	PFC IN area Volatile PFC input variables
8192...12287	0x2000...0x2FFF	%IX256.0...%IX511.15	PFC IN area Volatile PFC input variables
12288...32767	0x3000...0x7FFF	%MX0...%MX1279.15	NOVRAM 2 kB retain memory
32768...34295	0x8000...0x85F7	Physical output area (2)	Starts with the 513 th and ends with the 2039 th digital input
34296...36863	0x85F8...0x8FFF	-	MODBUS-Exception: "Illegal data address"
36864...38391	0x9000...0x95F7	Physical output area (2)	Starts with the 513 th and ends with the 2039 th digital output
38392...65535	0x95F8...0xFFFF	-	MODBUS-Exception: "Illegal data address"

12.2.5 MODBUS Registers

Table 157: MODBUS Registers

Register address	Access	Length (Word)	Description
0x1000	R/W	1	Watchdog time read/write
0x1001	R/W	1 ... 2	Watchdog coding mask 1...16
0x1002	R/W	1	Watchdog coding mask 17...32
0x1003	R/W	1	Watchdog trigger
0x1004	R	1	Minimum trigger time
0x1005	R/W	1	Watchdog stop (Write sequence 0xAAAA, 0x5555)
0x1006	R	1	Watchdog status
0x1007	R/W	1	Restart watchdog (Write sequence 0x1)
0x1008	R/W	1	Stop watchdog (Write sequence 0x55AA or 0xAA55)
0x1009	R/W	1	MODBUS and HTTP close at watchdog time-out
0x100A	R/W	1	Watchdog configuration
0x100B	W	1	Save watchdog parameter (Write sequence 0x55AA or 0xAA55)
0x1020	R	1 ... 2	LED error code
0x1021	R	1	LED error argument
0x1022	R	1 ... 4	Number of analog output data in the process image (in bits)
0x1023	R	1 ... 3	Number of analog input data in the process image (in bits)
0x1024	R	1 ... 2	Number of digital output data in the process image (in bits)
0x1025	R	1	Number of digital input data in the process image (in bits)
0x1028	R/W	1	Boot configuration
0x1029	R/W	1 ... 9	MODBUS/TCP statistics (Write sequence 0x55AA or 0xAA55)
0x102A	R	1	Number of TCP connections
0x102B	W	1	KBUS Reset
0x1030	R/W	1	Configuration MODBUS/TCP time-out
0x1031	R	1 ... 3	Read out the MAC-ID of the coupler/controller
0x1035	R/W	1	Timeoffset RTC
0x1036	R/W	1	Daylight Saving
0x1037	R/W	1	Modbus Response Delay (ms)
0x1050	R	3	Diagnosis of the connected I/O modules
0x2000	R	1 ... 9	Constant 0x0000
0x2001	R	1 ... 8	Constant 0xFFFF
0x2002	R	1 ... 7	Constant 0x1234
0x2003	R	1 ... 6	Constant 0xAAAA
0x2004	R	1 ... 5	Constant 0x5555
0x2005	R	1 ... 4	Constant 0x7FFF
0x2006	R	1 ... 3	Constant 0x8000
0x2007	R	1 ... 2	Constant 0x3FFF
0x2008	R	1	Constant 0x4000
0x2010	R	1	Firmware version
0x2011	R	1	Series code
0x2012	R	1	Coupler/controller code
0x2013	R	1	Firmware version major revision
0x2014	R	1	Firmware version minor revision

Table 158: MODBUS registers (Continuation)

Register address	Access	Length (Word)	Description
0x2020	R	1 ... 16	Short description controller
0x2021	R	1 ... 8	Compile time of the firmware
0x2022	R	1 ... 8	Compile date of the firmware
0x2023	R	1 ... 32	Indication of the firmware loader
0x2030	R	1 ... 65	Description of the connected I/O modules (module 0...64)
0x2031	R	1 ... 64	Description of the connected I/O modules (module 65...128)
0x2032	R	1 ... 64	Description of the connected I/O modules (module 129...192)
0x2033	R	1 ... 63	Description of the connected I/O modules (module 193...255)
0x2040	W	1	Software reset (Write sequence 0x55AA or 0xAA55)
0x2041	W	1	Format flash disk
0x2042	W	1	Extract HTML sides from the firmware
0x2043	W	1	Factory settings

12.2.5.1 Accessing Register Values

You can use any MODBUS application to access (read from or write to) register values. Both commercial (e.g., "Modscan") and free programs (from <http://www.modbus.org/tech.php>) are available.

The following sections describe how to access both the registers and their values.

12.2.5.2 Watchdog Registers

The watchdog monitors the data transfer between the fieldbus master and the controller. Every time the controller receives a specific request (as define in the watchdog setup registers) from the master, the watchdog timer in the controller resets.

In the case of fault free communication, the watchdog timer does not reach its end value. After each successful data transfer, the timer is reset.

If the watchdog times out, a fieldbus failure has occurred. In this case, the fieldbus controller answers all following MODBUS TCP/IP requests with the exception code 0x0004 (Slave Device Failure).

In the controller special registers are used to setup the watchdog by the master (Register addresses 0x1000 to 0x1008).

By default, the watchdog is not enabled when you turn the controller on. To activate it, the first step is to set/verify the desired time-out value of the Watchdog Time register (0x1000). Second, the function code mask must be specified in the mask register (0x1001), which defines the function code(s) that will reset the timer for the first time. Finally, the Watchdog-Trigger register (0x1003) or the register 0x1007 must be changed to a non-zero value to start the timer subsequently.

Reading the Minimum Trigger time (Register 0x1004) reveals whether a watchdog fault occurred. If this time value is 0, a fieldbus failure is assumed. The

timer of watchdog can manually be reset, if it is not timed out, by writing a value of 0x1 to the register 0x1003 or to the Restart Watchdog register 0x1007.

After the watchdog is started, it can be stopped by the user via the Watchdog Stop register (0x1005) or the Simply Stop Watchdog register (0x1008).

The watchdog registers can be addressed in the same way as described with the MODBUS read and write function codes. Specify the respective register address in place of the reference number.

Table 159: Register Address 0x1000

Register address 0x1000 (4096_{dec})	
Value	Watchdog time, WS_TIME
Access	Read/write
Default	0x0064
Description	This register stores the watchdog timeout value as an unsigned 16 bit value. The default value is 0. Setting this value will not trigger the watchdog. However, a non zero value must be stored in this register before the watchdog can be triggered. The time value is stored in multiples of 100ms (e.g., 0x0009 is .9 seconds). It is not possible to modify this value while the watchdog is running.

Table 160: Register Address 0x1001

Register address 0x1001 (4097_{dec})	
Value	Watchdog function coding mask, function code 1...16, WDFCM_1_16
Access	Read/write
Default	0xFFFF
Description	Using this mask, the function codes can be set to trigger the watchdog function. The function code can be selected via a "1" FC 1 Bit 0 FC 2 Bit 1 FC 3 Bit 2 FC 4 Bit 3 FC 5 Bit 4 ... FC 16 Bit 15 Changes to the register value can only be made if the watchdog is deactivated. The bit pattern stored in the register defines the function codes that trigger the watchdog. Some function codes are not supported. For those the watchdog will not be triggered even if another MODBUS device transmits one of them.

Table 161: Register Address 0x1002

Register address 0x1002 (4098 _{dec})	
Value	Watchdog function coding mask, function code 17...32, WD_FCM_17_32
Access	Read/write
Default	0xFFFF
Description	Same function as above, however, with the function codes 17 to 32. FC 17 Bit 0 FC 18 Bit 1 ... FC 32 Bit 15 These codes are currently not supported, for this reason the default value should not be changed. Changes to the register value can only be made if the watchdog is deactivated. It is not possible to modify this value while the watchdog is running.

Table 162: Register Address 0x1003

Register address 0x1003 (4099 _{dec})	
Value	Watchdog trigger, WD_TRIGGER
Access	Read/write
Standard	0x0000
Description	This register is used for an alternative trigger method. The watchdog is triggered by writing different values in this register. Values following each other must differ in size. Writing of a value not equal to zero starts the watchdog after a Power-on. For a restart the written value must necessarily be unequal the before written value! A watchdog fault is reset and writing process data is possible again.

Table 163: Register Address 0x1004

Register address 0x1004 (4100 _{dec})	
Value	Minimum current trigger time, WD_AC_TRG_TIME
Access	Read
Standard	0xFFFF
Description	This register saves the minimum current watchdog trigger time. If the watchdog is triggered, the saved value is compared with the current value. If the current value is smaller than the saved value, this is replaced by the current value. The unit is 100 ms/digit. The saved value is changed by writing new values, which does not affect the watchdog. 0x0000 is not permissible.

Table 164: Register Address 0x1005

Register address 0x1005 (4101 _{dec})	
Value	Stop watchdog, WD_AC_STOP_MASK
Access	Read/write
Standard	0x0000
Description	The watchdog is stopped if here the value 0xAAAA is written first, followed by 0x5555. The watchdog fault reaction is blocked. A watchdog fault is reset and writing on the process data is possible again.

Table 165: Register Address 0x1006

Register address 0x1006 (4102 _{dez})	
Value	While watchdog is running, WD_RUNNING
Access	Read
Standard	0x0000
Description	Current watchdog status. at 0x0000: Watchdog not active at 0x0001: Watchdog active at 0x0002: Watchdog exhausted.

Table 166: Register Address 0x1007

Register address 0x1007 (4103 _{dez})	
Value	Restart watchdog, WD_RESTART
Access	Read/write
Standard	0x0000
Description	This register restarts the watchdog timer by writing a value of 0x1 into it. If the watchdog was stopped before the overrun, it is not restarted.

Table 167: Register Address 0x1008

Register address 0x1008 (4104 _{dez})	
Value	Simply stop watchdog, WD_AC_STOP_SIMPLE
Access	Read/write
Standard	0x0000
Description	This register stops the watchdog by writing the value 0xAA55 or 0x55AA into it. The watchdog timeout fault is deactivated and it is possible to write in the watchdog register again. If there is an existing watchdog fault, it is reset

Table 168: Register Address 0x1009

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

Table 169: Register Address 0x100A

Register address 0x100A (4106 _{dez})	
Value	Alternative watchdog
Access	Read/write
Standard	0x0000
Description	This register provides an alternate way to activate the watchdog timer. Procedure: Write a time value in register 0x1000; then write a 0x0001 into register 0x100A. With the first MODBUS request, the watchdog is started. The watchdog timer is reset with each MODBUS/TCP instruction. If the watchdog times out, all outputs are set to zero. The outputs will become operational again, after communications are re-established. Register 0x00A is non-volatile, including register 0x1000. It is not possible to modify the time value in register 0x1000 while the watchdog is running.

The length of each register is 1 word; i.e., with each access only one word can be written or read. Following are two examples of how to set the value for a time overrun:

Setting the watchdog for a timeout of more than 1 second:

1. Write 0x000A in the register for time overrun (0x1000).
Register 0x1000 works with a multiple of 100 ms;
 $1\text{ s} = 1000\text{ ms}$; $1000\text{ ms} / 100\text{ ms} = 10_{\text{dec}} = A_{\text{hex}}$
2. Use the function code 5 to write $0x0010 (=2^{(5-1)})$ in the coding mask (register 0x1001).

Table 170: Starting Watchdog

FC	FC16	FC15	FC14	FC13	FC12	FC11	FC10	FC9	FC8	FC7	FC6	FC5	FC4	FC3	FC2	FC1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bin	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
hex	0				0				1				0			

Function code 5 (writing a digital output bit) continuously triggers the watchdog to restart the watchdog timer again and again within the specified time. If time between requests exceeds 1 second, a watchdog timeout error occurs.

3. To stop the watchdog, write the value 0xAA55 or 0x55AA into 0x1008 (Simply Stop Watchdog register, WD_AC_STOP_SIMPLE).

Setting the watchdog for a timeout of 10 minutes or more:

1. Write 0x1770 ($= 10 \cdot 60 \cdot 1000\text{ ms} / 100\text{ ms}$) in the register for time overrun (0x1000).
(Register 0x1000 works with a multiple of 100 ms;
 $10\text{ min} = 600,000\text{ ms}$; $600,000\text{ ms} / 100\text{ ms} = 6000_{\text{dec}} = 1770_{\text{hex}}$)
2. Write 0x0001 in the watchdog trigger register (0x1003) to start the watchdog.
3. Write different values (e.g., counter values 0x0000, 0x0001) in the watchdog to trigger register (0x1003).

Values following each other must differ in size. Writing of a value not equal to zero starts the watchdog. Watchdog faults are reset and writing process data is possible again.

4. To stop the watchdog, write the value 0xAA55 or 0x55AA into 0x1008 (Simply Stop Watchdog register, WD_AC_STOP_SIMPLE).

Table 171: Register Address 0x100B

Register address 0x100B (4107 _{dez})	
Value	Save watchdog parameter
Access	Write
Standard	0x0000
Description	With writing of '0x55AA' or '0xAA55' in register 0x100B the registers 0x1000, 0x1001, 0x1002 are set on remanent.

12.2.5.3 Diagnostic Registers

The following registers can be read to determine errors in the node:

Table 172: Register Address 0x1020

Register address 0x1020 (4128 _{dec})	
Value	LedErrCode
Access	Read
Description	Declaration of the error code

Table 173: Register Address 0x1021

Register address 0x1021 (4129 _{dec})	
Value	LedErrArg
Access	Read
Description	Declaration of the error argument

12.2.5.4 Configuration Registers

The following registers contain configuration information of the connected modules:

Table 174: Register Address 0x1022

Register address 0x1022 (4130 _{dec})	
Value	CnfLen.AnalogOut
Access	Read
Description	Number of word-based outputs registers in the process image in bits (divide by 16 to get the total number of analog words)

Table 175: Register Address 0x1023

Register address 0x1023 (4131 _{dec})	
Value	CnfLen.AnalogInp
Access	Read
Description	Number of word-based inputs registers in the process image in bits (divide by 16 to get the total number of analog words)

Table 176: Register Address 0x1024

Register address 0x1024 (4132 _{dec})	
Value	CnfLen.DigitalOut
Access	Read
Description	Number of digital output bits in the process image

Table 177: Register Address 0x1025

Register address 0x1025 (4133 _{dec})	
Value	CnfLen.DigitalInp
Access	Read
Description	Number of digital input bits in the process image

Table 178: Register Address 0x1028

Register address 0x1028 (4136 _{dec})	
Value	Boot options
Access	Read/write
Description	Boot configuration: 1: BootP 2: DHCP 3: BootP-Request before static IP 4: EEPROM

Table 179: Register Address 0x1029

Register address 0x1029 (4137 _{dec}) with 9 words		
Value	MODBUS TCP statistics	
Access	Read/write	
Description	1 word SlaveDeviceFailure	→ internal bus error, fieldbus error by activated watchdog
	1 word BadProtocol	→ error in the MODBUS TCP header
	1 word BadLength	→ Wrong telegram length
	1 word BadFunction	→ Invalid function code
	1 word BadAddress	→ Invalid register address
	1 word BadData	→ Invalid value
	1 word TooManyRegisters	→ Number of the registers which can be worked on is too large, Read/Write 125/100
	1 word TooManyBits	→ Number of the coils which can be worked on is too large, Read/Write 2000/800
	1 word ModTcpMessageCounter	→ Number of received MODBUS/TCP requests
With writing 0xAA55 or 0x55AA in the register will reset this data area.		

Table 180: Register Address 0x102A

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

Table 181: Register Address 0x102B

Register address 0x102B (4139 _{dec}) with a word count of up to 1	
Value	KBUS reset
Access	Write
Description	Writing of this register restarts the internal bus

Table 182: Register Address 0x1030

Register address 0x1030 (4144 _{dec}) with a word count of 1	
Value	Configuration MODBUS/TCP time-out
Access	Read/write
Default	0x0258 (600 decimal)
Description	This is the maximum number of milliseconds the fieldbus coupler will allow a MODBUS/TCP connection to stay open without receiving a MODBUS request. Upon time-out, idle connection will be closed. Outputs remain in last state. Default value is 600 ms (60 seconds), the time base is 100 ms, the minimal value is 100 ms. If the value is set to '0', the timeout is disabled. On this connection, the watchdog is triggered with a request.

Table 183: Register Address 0x1031

Register address 0x1031 (4145 _{dec}) with a word count of 3	
Value	Read the MAC-ID of the controller
Access	Read
Description	This register gives the MAC-ID, with a length of 3 words

Table 184: Register Address 0x1035

Register address 0x1035 (4149 _{dez}) 1 word	
Value	Configuration of the time offsets to the GMT time
Access	Read/write
Default	0x0000
Description	Register to set the time offset to the UTC time (Greenwich meridian) with a possible setting range from -12 to +12.

Table 185: Register Address 0x1036

Register address 0x1036 (4150 _{dez}) 1 word	
Value	Configuration of summer or winter time
Access	Read/write
Default	0x0000
Description	Register to set winter or summer time (Daylight Saving Time). The values 0 and 1 are valid.

Table 186: Register Address 0x1037

Register address 0x1031 (4151 _{dez}) with a word count of 3	
Value	Configuration of Modbus Response Delay Time
Access	Read/write
Default	0x0000
Description	This register saves the value for the Modbus Response Delay Time for a Modbus connection. The time base is 1 ms. On the Modbus TCP connection, the response will be delayed by the inscribed time.

Table 187: Register Address 0x1050

Register address 0x1050 (4176 _{dec}) with a word count of 3	
Value	Diagnosis of the connected I/O modules
Access	Read
Description	Diagnosis of the connected I/O modules, length 3 words Word 1: Number of the module Word 2: Number of the channel Word 3: Diagnosis

Table 188: Register Address 0x2030

Register address 0x2030 (8240 _{dec}) with a word count of up to 65																
Value	Description of the connected I/O modules															
Access	Read module 0...64															
Description	Length 1...65 words															
	These 65 registers identify the controller and the first 64 modules present in a node. Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below:															
	Bit position 0				→		Input module									
	Bit position 1				→		Output module									
	Bit position 2...7				→		Not used									
	Bit position 8...14				→		Module size in bits									
	Bit position 15				→		Designation digital module									
	Examples:															
	4 Channel Digital Input Module = 0x8401															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Code	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
Hex	8				4				0				1			
2 Channel Digital Output Module = 0x8202																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Code	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
Hex	8				2				0				2			

Table 189: Register Address 0x2031

Register address 0x2031 (8241 _{dec}) with a word count of up to 64															
Value	Description of the connected I/O modules														
Access	Read modules 65...128														
Description	Length 1...64 words														
	These 64 registers identify the 2nd block of I/O modules present (modules 65 to 128). Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below:														
	Bit position 0	→		Input module											
	Bit position 1	→		Output module											
	Bit position 2...7	→		Not used											
	Bit position 8...14	→		Module size in bits											
	Bit position 15	→		Designation digital module											

Table 190: Register Address 0x2032

Register address 0x2032 (8242 _{dec}) with a word count of up to 64															
Value	Description of the connected I/O modules														
Access	Read modules 129...192														
Description	Length 1...64 words														
	These 64 registers identify the 3rd block of I/O modules present (modules 129 to 192). Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below:														
	Bit position 0	→		Input module											
	Bit position 1	→		Output module											
	Bit position 2...7	→		Not used											
	Bit position 8...14	→		Module size in bits											
	Bit position 15	→		Designation digital module											

Table 191: Register Address 0x2033

Register address 0x2033 (8243 _{dec}) with a word count of up to 65	
Value	Description of the connected I/O modules
Access	Read modules 193 ... 255
Description	Length 1...63 words These 63 registers identify the 4th block of I/O modules present (modules 193 to 255). Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 → Input module Bit position 1 → Output module Bit position 2...7 → Not used Bit position 8...14 → Module size in bits Bit position 15 → Designation digital module

Table 192: Register Address 0x2040

Register address 0x2040 (8256 _{dec})	
Value	Implement a software reset
Access	Write (Write sequence 0xAA55 or 0x55AA)
Description	With writing 0xAA55 or 0x55AA the register will be reset.

Table 193: Register Address 0x2041

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

Table 194: Register Address 0x2042

Register address 0x2042 (8258 _{dez})	
Value	Extract data files
Access	Write (Write sequence 0xAA55 or 0x55AA)
Description	The standard files (HTML pages) of the Coupler/Controller are extracted and written into the Flash.

Table 195: Register Address 0x2043

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

12.2.5.5 Firmware Information Registers

The following registers contain information on the firmware of the controller:

Table 196: Register Address 0x2010

Register address 0x2010 (8208 _{dec}) with a word count of 1	
Value	Revision, INFO_REVISION
Access	Read
Description	Firmware index, e.g. 0x0005 for version 5

Table 197: Register Address 0x2011

Register address 0x2011 (8209 _{dec}) with a word count of 1	
Value	Series code, INFO_SERIES
Access	Read
Description	WAGO serial number, e.g. 0x02EE (750 dec.) for WAGO-I/O-SYSTEM 750

Table 198: Register Address 0x2012

Register address 0x2012 (8210 _{dec}) with a word count of 1	
Value	Order number, INFO_ITEM
Access	Read
Description	First part of WAGO order number, e.g. 0x0349 (841 dec.) for the controller 750-841 or 0x0155 (341 dec.) for the coupler 750-341 etc.

Table 199: Register Address 0x2013

Register address 0x2013 (8211 _{dec}) with a word count of 1	
Value	Major sub item code, INFO_MAJOR
Access	Read
Description	Firmware version major revision

Table 200: Register Address 0x2014

Register address 0x2014 (8212 _{dec}) with a word count of 1	
Value	Minor sub item code, INFO_MINOR
Access	Read
Description	Firmware version minor revision

Table 201: Register Address 0x2020

Register address 0x2020 (8224 _{dec}) with a word count of up to 16	
Value	Description, INFO_DESCRIPTION
Access	Read
Description	Information on the controller, 16 words

Table 202: Register Address 0x2021

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

Table 203: Register Address 0x2022

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

Table 204: Register Address 0x2023

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

12.2.5.6 Constant Registers

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

Table 205: Register Address 0x2000

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

Table 206: Register Address 0x2001

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

Table 207: Register Address 0x2002

Register address 0x2002 (8194 _{dec})	
Value	1,2,3,4, GP_1234
Access	Read
Description	This constant value is used to test the Intel/Motorola format specifier. If the master reads a value of 0x1234, then with Intel format is selected – this is the correct format. If 0x3412 appears, Motorola format is selected.

Table 208: Register Address 0x2003

Register address 0x2003 (8195 _{dec})	
Value	Mask 1, GP_AAAA
Access	Read
Description	This constant is used to verify that all bits are accessible to the fieldbus master. This will be used together with register 0x2004.

Table 209: Register Address 0x2004

Register address 0x2004 (8196 _{dec})	
Value	Mask 1, GP_5555
Access	Read
Description	This constant is used to verify that all bits are accessible to the fieldbus master. This will be used together with register 0x2003.

Table 210: Register Address 0x2005

Register address 0x2005 (8197 _{dec})	
Value	Maximum positive number, GP_MAX_POS
Access	Read
Description	Constant in order to control arithmetic.

Table 211: Register Address 0x2006

Register address 0x2006 (8198 _{dec})	
Value	Maximum negative number, GP_MAX_NEG
Access	Read
Description	Constant in order to control arithmetic

Table 212: Register Address 0x2007

Register address 0x2007 (8199 _{dec})	
Value	Maximum half positive number, GP_HALF_POS
Access	Read
Description	Constant in order to control arithmetic

Table 213: Register Address 0x2008

Register address 0x2008 (8200 _{dec})	
Value	Maximum half negative number, GP_HALF_NEG
Access	Read
Description	Constant in order to control arithmetic

Table 214: Register Addresses 0x3000 to 0x5FFF

Register addresses 0x3000 to 0x5FFF (12288 _{dec} to 24575 _{dec})	
Value	Retain range
Access	Read/write
Description	These registers can be accessed as the flag/retain range

13 I/O Modules

13.1 Overview

For modular applications with the WAGO-I/O-SYSTEM 750, different types of I/O modules are available

- Digital Input Modules
- Digital Output Modules
- Analog Input Modules
- Analog Output Modules
- Specialty Modules
- System Modules

For detailed information on the I/O modules and the module variations, refer to the manuals for the I/O modules.

You will find these manuals on the WAGO web pages under www.wago.com.



Information

More Information about the WAGO-I/O-SYSTEM

Current information on the modular WAGO-I/O-SYSTEM is available in the Internet under: www.wago.com.

13.2 Process Data Architecture for MODBUS/TCP

With some I/O modules, the structure of the process data is fieldbus specific.

MODBUS/TCP process image uses a word structure (with word alignment). The internal mapping method for data greater than one byte conforms to the Intel format.

The following section describes the process image for various WAGO-I/O-SYSTEM 750 and 753 I/O modules with MODBUS/TCP.

NOTICE

Equipment damage due to incorrect address!

Depending on the specific position of an I/O module in the fieldbus node, the process data of all previous byte or bit-oriented modules must be taken into account to determine its location in the process data map.

The structure of the process data mapping is identical for the PFC process image of the programmable fieldbus controller.

13.2.1 Digital Input Modules

Digital input modules supply one bit of data per channel to specify the signal state for the corresponding channel. These bits are mapped into the Input Process Image.

Some digital modules have an additional diagnostic bit per channel in the Input Process Image. The diagnostic bit is used for detecting faults that occur (e.g., wire breaks and/or short circuits).

When analog input modules are also present in the node, the digital data is always appended after the analog data in the Input Process Image, grouped into bytes.

13.2.1.1 1 Channel Digital Input Module with Diagnostics

750-435

Table 215: 1 Channel Digital Input Module with Diagnostics

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 1	Data bit DI 1

13.2.1.2 2 Channel Digital Input Modules

750-400, -401, -405, -406, -410, -411, -412, -427, -438, (and all variations),
753-400, -401, -405, -406, -410, -411, -412, -427

Table 216: 2 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.3 2 Channel Digital Input Module with Diagnostics

750-419, -421, -424, -425,
753-421, -424, -425

Table 217: 2 Channel Digital Input Module with Diagnostics

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.4 2 Channel Digital Input Module with Diagnostics and Output Process Data

750-418,
753-418

The digital input module supplies a diagnostic and acknowledge bit for each input channel. If a fault condition occurs, the diagnostic bit is set. After the fault condition is cleared, an acknowledge bit must be set to re-activate the input. The diagnostic data and input data bit is mapped in the Input Process Image, while the acknowledge bit is in the Output Process Image.

Table 218: 2 Channel Digital Input Module with Diagnostics and Output Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Acknowledge- ment bit Q 2 Channel 2	Acknowledge- ment bit Q 1 Channel 1	0	0

13.2.1.5 4 Channel Digital Input Modules

750-402, -403, -408, -409, -414, -415, -422, -423, -428, -432, -433, -1420, -1421,
-1422, -1423
753-402, -403, -408, -409, -415, -422, -423, -428, -432, -433, -440

Table 219: 4 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.6 8 Channel Digital Input Modules

750-430, -431, -436, -437, -1415, -1416, -1417, -1418
753-430, -431, -434

Table 220: 8 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.7 8 Channel Digital Input Module PTC with Diagnostics and Output Process Data

750-1425

The digital input module PTC provides via one logical channel 2 byte for the input and output process image.

The signal state of PTC inputs DI1 ... DI8 is transmitted to the fieldbus coupler/controller via input data byte D0.

The fault conditions are transmitted via input data byte D1.

The channels 1 ... 8 are switched on or off via the output data byte D1. The output data byte D0 is reserved and always has the value "0".

Table 221: 8 Channel Digital Input Module PTC with Diagnostics and Output Process Data

Input Process Image															
Input Byte D0								Input Byte D1							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Signal status DI 8 Channel 8	Signal status DI 7 Channel 7	Signal status DI 6 Channel 6	Signal status DI 5 Channel 5	Signal status DI 4 Channel 4	Signal status DI 3 Channel 3	Signal status DI 2 Channel 2	Signal status DI 1 Channel 1	Wire break/short circuit DB/KS 8 Channel 8	Wire break/short circuit DB/KS 7 Channel 7	Wire break/short circuit DB/KS 6 Channel 6	Wire break/short circuit DB/KS 5 Channel 5	Wire break/short circuit DB/KS 4 Channel 4	Wire break/short circuit DB/KS 3 Channel 3	Wire break/short circuit DB/KS 2 Channel 2	Wire break/short circuit DB/KS 1 Channel 1
Output Process Image															
Output Byte D0								Output Byte D1							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0	DI Off 8 Channel 8 0: Channel ON 1: Channel OFF	DI Off 7 Channel 7 0: Channel ON 1: Channel OFF	DI Off 6 Channel 6 0: Channel ON 1: Channel OFF	DI Off 5 Channel 5 0: Channel ON 1: Channel OFF	DI Off 4 Channel 4 0: Channel ON 1: Channel OFF	DI Off 3 Channel 3 0: Channel ON 1: Channel OFF	DI Off 2 Channel 2 0: Channel ON 1: Channel OFF	DI Off 1 Channel 1 0: Channel ON 1: Channel OFF

13.2.1.8 16 Channel Digital Input Modules

750-1400, -1402, -1405, -1406, -1407

Table 222: 16 Channel Digital Input Modules

Input Process Image															
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 16 Channel 16	Data bit DI 15 Channel 15	Data bit DI 14 Channel 14	Data bit DI 13 Channel 13	Data bit DI 12 Channel 12	Data bit DI 11 Channel 11	Data bit DI 10 Channel 10	Data bit DI 9 Channel 9	Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.2 Digital Output Modules

Digital output modules use one bit of data per channel to control the output of the corresponding channel. These bits are mapped into the Output Process Image.

Some digital modules have an additional diagnostic bit per channel in the Input Process Image. The diagnostic bit is used for detecting faults that occur (e.g., wire breaks and/or short circuits). For modules with diagnostic bit is set, also the data bits have to be evaluated.

When analog output modules are also present in the node, the digital image data is always appended after the analog data in the Output Process Image, grouped into bytes.

13.2.2.1 1 Channel Digital Output Module with Input Process Data

750-523

The digital output modules deliver 1 bit via a process value Bit in the output process image, which is illustrated in the input process image. This status image shows "manual mode".

Table 223: 1 Channel Digital Output Module with Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						not used	Status bit "Manual Operation"

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						not used	controls DO 1 Channel 1

13.2.2.2 2 Channel Digital Output Modules

750-501, -502, -509, -512, -513, -514, -517, -535, (and all variations),
753-501, -502, -509, -512, -513, -514, -517

Table 224: 2 Channel Digital Output Modules

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.3 2 Channel Digital Input Modules with Diagnostics and Input Process Data

750-507 (-508), -522,
753-507

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 225: 2 Channel Digital Input Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						controls DO 2 Channel 2	controls DO 1 Channel 1

750-506,
753-506

The digital output module has 2-bits of diagnostic information for each output channel. The 2-bit diagnostic information can then be decoded to determine the exact fault condition of the module (i.e., overload, a short circuit, or a broken wire). The 4-bits of diagnostic data are mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 226: 2 Channel Digital Input Modules with Diagnostics and Input Process Data 75x-506

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 3 Channel 2	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Diagnostic bit S 0 Channel 1

Diagnostic bits S1/S0, S3/S2: = '00'
Diagnostic bits S1/S0, S3/S2: = '01'
Diagnostic bits S1/S0, S3/S2: = '10'

standard mode
no connected load/short circuit against +24 V
Short circuit to ground/overload

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				not used	not used	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.4 4 Channel Digital Output Modules

750-504, -516, -519, -531,
753-504, -516, -531, -540

Table 227: 4 Channel Digital Output Modules

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.5 4 Channel Digital Output Modules with Diagnostics and Input Process Data

750-532

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 228: 4 Channel Digital Output Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 4 Channel 4	Diagnostic bit S 3 Channel 3	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Diagnostic bit S = '0'

no Error

Diagnostic bit S = '1'

overload, short circuit, or broken wire

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.6 8 Channel Digital Output Module

750-530, -536, -1515, -1516
753-530, -534

Table 229: 8 Channel Digital Output Module

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.7 8 Channel Digital Output Modules with Diagnostics and Input Process Data

750-537

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 230: 8 Channel Digital Output Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Diagnostic bit S 8 Channel 8	Diagnostic bit S 7 Channel 7	Diagnostic bit S 6 Channel 6	Diagnostic bit S 5 Channel 5	Diagnostic bit S 4 Channel 4	Diagnostic bit S 3 Channel 3	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Diagnostic bit S = '0' no Error
Diagnostic bit S = '1' overload, short circuit, or broken wire

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.8 16 Channel Digital Output Modules

750-1500, -1501, -1504, -1505

Table 231: 16 Channel Digital Output Modules

Output Process Image															
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 16 Channel 16	controls DO 15 Channel 15	controls DO 14 Channel 14	controls DO 13 Channel 13	controls DO 12 Channel 12	controls DO 11 Channel 11	controls DO 10 Channel 10	controls DO 9 Channel 9	controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.9 8 Channel Digital Input/Output Modules

750-1502, -1506

Table 232: 8 Channel Digital Input/Output Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 8	Data bit DI 7	Data bit DI 6	Data bit DI 5	Data bit DI 4	Data bit DI 3	Data bit DI 2	Data bit DI 1
Channel 8	Channel 7	Channel 6	Channel 5	Channel 4	Channel 3	Channel 2	Channel 1

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 8	controls DO 7	controls DO 6	controls DO 5	controls DO 4	controls DO 3	controls DO 2	controls DO 1
Channel 8	Channel 7	Channel 6	Channel 5	Channel 4	Channel 3	Channel 2	Channel 1

13.2.3 Analog Input Modules

The hardware of an analog input module has 16 bits of measured analog data per channel and 8 bits of control/status.

However, the coupler/controller with MODBUS/TCP does not have access to the 8 control/status bits.

Therefore, the coupler/controller with MODBUS/TCP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Input Process Image.

When digital input modules are also present in the node, the analog input data is always mapped into the Input Process Image in front of the digital data.



Information

Information on the structure of control and status bytes

For detailed information on the structure of a particular I/O module's control/status bytes, please refer to that module's manual. Manuals for each module can be found on the Internet at www.wago.com.

13.2.3.1 1 Channel Analog Input Modules

750-491, (and all variations)

Table 233: 1 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Measured Value U_D
1	D3	D2	Measured Value U_{ref}

13.2.3.2 2 Channel Analog Input Modules

750-452, -454, -456, -461, -462, -465, -466, -467, -469, -472, -474, -475, 476, -477, -478, -479, -480, -481, -483, -485, -492, (and all variations),
753-452, -454, -456, -461, -465, -466, -467, -469, -472, -474, -475, 476, -477, 478, -479, -483, -492, (and all variations)

Table 234: 2 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Measured Value Channel 1
1	D3	D2	Measured Value Channel 2

13.2.3.3 4 Channel Analog Input Modules

750-450, -453, -455, -457, -459, -460, -468, (and all variations),
753-453, -455, -457, -459

Table 235: 4 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Measured Value Channel 1
1	D3	D2	Measured Value Channel 2
2	D5	D4	Measured Value Channel 3
3	D7	D6	Measured Value Channel 4

13.2.3.4 3-Phase Power Measurement Module

750-493

The above Analog Input Modules have a total of 9 bytes of user data in both the Input and Output Process Image (6 bytes of data and 3 bytes of control/status). The following tables illustrate the Input and Output Process Image, which has a total of 6 words mapped into each image. Word alignment is applied.

Table 236: 3-Phase Power Measurement Module

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	S0	Status byte 0
1	D1	D0	Input data word 1
2	-	S1	Status byte 1
3	D3	D2	Input data word 2
4	-	S2	Status byte 2
5	D5	D4	Input data word 3

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C0	Control byte 0
1	D1	D0	Output data word 1
2	-	C1	Control byte 1
3	D3	D2	Output data word 2
4	-	C2	Control byte 2
5	D5	D4	Output data word 3

13.2.3.5 8 Channel Analog Input Modules

750-451

Table 237: 8 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Measured Value Channel 1
1	D3	D2	Measured Value Channel 2
2	D5	D4	Measured Value Channel 3
3	D7	D6	Measured Value Channel 4
4	D9	D8	Measured Value Channel 5
5	D11	D10	Measured Value Channel 6
6	D13	D12	Measured Value Channel 7
7	D15	D14	Measured Value Channel 8

13.2.4 Analog Output Modules

The hardware of an analog output module has 16 bits of measured analog data per channel and 8 bits of control/status. However, the coupler/controller with MODBUS/TCP does not have access to the 8 control/status bits. Therefore, the coupler/controller with MODBUS/TCP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Output Process Image.

When digital output modules are also present in the node, the analog output data is always mapped into the Output Process Image in front of the digital data.



Information

Information on the structure of control and status bytes

For detailed information on the structure of a particular I/O module's control/status bytes, please refer to that module's manual. Manuals for each module can be found on the Internet at www.wago.com.

13.2.4.1 2 Channel Analog Output Modules

750-550, -552, -554, -556, -560, -562, 563, -585, (and all variations),
753-550, -552, -554, -556

Table 238: 2 Channel Analog Output Modules

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Output Value Channel 1
1	D3	D2	Output Value Channel 2

13.2.4.2 4 Channel Analog Output Modules

750-553, -555, -557, -559,
753-553, -555, -557, -559

Table 239: 4 Channel Analog Output Modules

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Output Value Channel 1
1	D3	D2	Output Value Channel 2
2	D5	D4	Output Value Channel 3
3	D7	D6	Output Value Channel 4

13.2.4.3 8 Channel Analog Output Modules

Table 240: 8 Channel Analog Output Modules

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Output Value Channel 1
1	D3	D2	Output Value Channel 2
2	D5	D4	Output Value Channel 3
3	D7	D6	Output Value Channel 4
4	D9	D8	Output Value Channel 5
5	D11	D10	Output Value Channel 6
6	D13	D12	Output Value Channel 7
7	D15	D14	Output Value Channel 8

13.2.5 Specialty Modules

WAGO has a host of Specialty I/O modules that perform various functions. With individual modules beside the data bytes also the control/status byte is mapped in the process image.

The control/status byte is required for the bidirectional data exchange of the module with the higher-ranking control system. The control byte is transmitted from the control system to the module and the status byte from the module to the control system.

This allows, for example, setting of a counter with the control byte or displaying of overshooting or undershooting of the range with the status byte.

The control/status byte always is in the process image in the Low byte.



Information

Information to the structure of the Control/Status byte

For detailed information about the structure of a particular module's control/status byte, please refer to that module's manual. Manuals for each module can be found on the Internet under: www.wago.com.

13.2.5.1 Counter Modules

750-404, (and all variations except of /000-005),
753-404, (and variation /000-003)

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/status). The counter value is supplied as 32 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

Table 241: Counter Modules 750-404, (and all variations except of /000-005),
753-404, (and variation /000-003)

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	S	Status byte
1	D1	D0	Counter value
2	D3	D2	

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C	Control byte
1	D1	D0	Counter setting value
2	D3	D2	

750-404/000-005

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/ status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

Table 242: Counter Modules 750-404/000-005

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	S	Status byte
1	D1	D0	Counter Value of Counter 1
2	D3	D2	Counter Value of Counter 2

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C	Control byte
1	D1	D0	Counter Setting Value of Counter 1
2	D3	D2	Counter Setting Value of Counter 2

750-638,
753-638

The above Counter Modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 2 bytes of control/status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

Table 243: Counter Modules 750-638, 753-638

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	S0	Status byte von Counter 1
1	D1	D0	Counter Value von Counter 1
2	-	S1	Status byte von Counter 2
3	D3	D2	Counter Value von Counter 2

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C0	Control byte von Counter 1
1	D1	D0	Counter Setting Value von Counter 1
2	-	C1	Control byte von Counter 2
3	D3	D2	Counter Setting Value von Counter 2

13.2.5.2 Pulse Width Modules

750-511, (and all variations /xxx-xxx)

The above Pulse Width modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of channel data and 2 bytes of control/status). The two channel values are supplied as 16 bits. Each channel has its own control/status byte. The following table illustrates the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

Table 244: Pulse Width Modules 750-511, /xxx-xxx

Input and Output Process			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C0/S0	Control/Status byte of Channel 1
1	D1	D0	Data Value of Channel 1
2	-	C1/S1	Control/Status byte of Channel 2
3	D3	D2	Data Value of Channel 2

13.2.5.3 Serial Interface Modules with alternative Data Format

750-650, (and the variations /000-002, -004, -006, -009, -010, -011, -012, -013),
750-651, (and the variations /000-001, -002, -003),
750-653, (and the variations /000-002, -007),
753-650, -653



Note

The process image of the / 003-000-variants depends on the parameterized operating mode!

With the freely parameterizable variations /003 000 of the serial interface modules, the desired operation mode can be set. Dependent on it, the process image of these modules is then the same, as from the appropriate variation.

The above Serial Interface Modules with alternative data format have a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 2 words mapped into each image. Word alignment is applied.

Table 245: Serial Interface Modules with alternative Data Format

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	C/S	Data byte	Control/status byte
1	D2	D1	Data bytes	

13.2.5.4 Serial Interface Modules with Standard Data Format

750-650/000-001, -014, -015, -016
750-653/000-001, -006

The above Serial Interface Modules with Standard Data Format have a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 3 words mapped into each image. Word alignment is applied.

Table 246: Serial Interface Modules with Standard Data Format

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	C/S	Data byte	Control/status byte
1	D2	D1	Data bytes	
2	D4	D3		

13.2.5.5 Data Exchange Module

750-654, (and the variation /000-001)

The Data Exchange modules have a total of 4 bytes of user data in both the Input and Output Process Image. The following tables illustrate the Input and Output Process Image, which has a total of 2 words mapped into each image. Word alignment is applied.

Table 247: Data Exchange Module

Input and Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Data bytes
1	D3	D2	

13.2.5.6 SSI Transmitter Interface Modules

750-630 (and all variations)



Note

The process image of the / 003-000-variants depends on the parameterized operating mode!

The operating mode of the configurable /003-000 I/O module versions can be set. Based on the operating mode, the process image of these I/O modules is then the same as that of the respective version.

The above SSI Transmitter Interface modules have a total of 4 bytes of user data in the Input Process Image, which has 2 words mapped into the image. Word alignment is applied.

Table 248: SSI Transmitter Interface Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Data bytes
1	D3	D2	

13.2.5.7 Incremental Encoder Interface Modules

750-631/000-004, -010, -011

The above Incremental Encoder Interface modules have 5 bytes of input data and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which have 4 words into each image. Word alignment is applied.

Table 249: Incremental Encoder Interface Modules 750-631/000-004, --010, -011

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	S	not used	Status byte
1	D1	D0	Counter word	
2	-	-	not used	
3	D4	D3	Latch word	

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C	not used	Control byte
1	D1	D0	Counter setting word	
2	-	-	not used	
3	-	-	not used	

750-634

The above Incremental Encoder Interface module has 5 bytes of input data (6 bytes in cycle duration measurement mode) and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which has 4 words mapped into each image. Word alignment is applied.

Table 250: Incremental Encoder Interface Modules 750-634

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	S	not used	Status byte
1	D1	D0	Counter word	
2	-	(D2) *)	not used	(Periodic time)
3	D4	D3	Latch word	

*) If cycle duration measurement mode is enabled in the control byte, the cycle duration is given as a 24-bit value that is stored in D2 together with D3/D4.

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C	not used	Control byte
1	D1	D0	Counter setting word	
2	-	-	not used	
3	-	-		

750-637

The above Incremental Encoder Interface Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of encoder data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Table 251: Incremental Encoder Interface Modules 750-637

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C0/S0	Control/Status byte of Channel 1	
1	D1	D0	Data Value of Channel 1	
2	-	C1/S1	Control/Status byte of Channel 2	
3	D3	D2	Data Value of Channel 2	

750-635,
753-635

The above Digital Pulse Interface module has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

Table 252: Digital Pulse Interface Modules 750-635

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	C0/S0	Data byte	Control/status byte
1	D2	D1	Data bytes	

13.2.5.8 DC-Drive Controller

750-636

The DC-Drive Controller maps 6 bytes into both the input and output process image. The data sent and received are stored in up to 4 input and output bytes (D0 ... D3). Two control bytes (C0, C1) and two status bytes (S0/S1) are used to control the I/O module and the drive.

In addition to the position data in the input process image (D0 ... D3), it is possible to display extended status information (S2 ... S5). Then the three control bytes (C1 ... C3) and status bytes (S1 ... S3) are used to control the data flow.

Bit 3 of control byte C1 (C1.3) is used to switch between the process data and the extended status bytes in the input process image (Extended Info_ON). Bit 3 of status byte S1 (S1.3) is used to acknowledge the switching process.

Table 253: DC-Drive Controller 750-636

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	S1	S0	Status byte S1	Status byte S0
1	D1*) / S3**)	D0*) / S2**)	Actual position*) / Extended status byte S3**)	Actual position (LSB) / Extended status byte S2**)
2	D3*) / S5**)	D2*) / S4**)	Actual position (MSB) / Extended status byte S3**)	Actual position*) / Extended status byte S4**)

*) ExtendedInfo_ON = '0'.

**) ExtendedInfo_ON = '1'.

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	C1	C0	Control byte C1	Control byte C0
1	D1	D0	Setpoint position	Setpoint position (LSB)
2	D3	D2	Setpoint position (MSB)	Setpoint position

13.2.5.9 Stepper Controller

750-670

The Stepper controller RS422 / 24 V / 20 mA 750-670 provides the fieldbus coupler 12 bytes input and output process image via 1 logical channel. The data to be sent and received are stored in up to 7 output bytes (D0 ... D6) and 7 input bytes (D0 ... D6), depending on the operating mode.

Output byte D0 and input byte D0 are reserved and have no function assigned.

One I/O module control and status byte (C0, S0) and 3 application control and status bytes (C1 ... C3, S1 ... S3) provide the control of the data flow.

Switching between the two process images is conducted through bit 5 in the control byte (C0 (C0.5). Activation of the mailbox is acknowledged by bit 5 of the status byte S0 (S0.5).

Table 254: Stepper Controller RS 422 / 24 V / 20 mA 750-670

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	reserved	S0	reserved	Status byte S0
1	D1	D0	Process data*) / Mailbox**)	
2	D3	D2		
3	D5	D4		
4	S3	D6	Status byte S3	Process data*) / reserved**)
5	S1	S2	Status byte S1	Status byte S2

*) Cyclic process image (Mailbox disabled)

**) Mailbox process image (Mailbox activated)

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	reserved	C0	reserved	Control byte C0
1	D1	D0	Process data*) / Mailbox**)	
2	D3	D2		
3	D5	D4		
4	C3	D6	Control byte C3	Process data*) / reserved**)
5	C1	C2	Control byte C1	Control byte C2

*) Cyclic process image (Mailbox disabled)

**) Mailbox process image (Mailbox activated)

13.2.5.10 RTC Module

750-640

The RTC Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of module data and 1 byte of control/status and 1 byte ID for command). The following table illustrates the Input and Output Process Image, which have 3 words mapped into each image. Word alignment is applied.

Table 255: RTC Module 750-640

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	ID	C/S	Command byte	Control/status byte
1	D1	D0	Data bytes	
2	D3	D2		

13.2.5.11 DALI/DSI Master Module

750-641

The DALI/DSI Master module has a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of module data and 1 byte of control/status). The following tables illustrate the Input and Output Process Image, which have 3 words mapped into each image. Word alignment is applied.

Table 256: DALI/DSI Master Module 750-641

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	S	DALI Response	Status byte
1	D2	D1	Message 3	DALI Address
2	D4	D3	Message 1	Message 2

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	C	DALI command, DSI dimming value	Control byte
1	D2	D1	Parameter 2	DALI Address
2	D4	D3	Command extension	Parameter 1

13.2.5.12 DALI Multi-Master Module

753-647

The DALI Multi-Master module occupies a total of 24 bytes in the input and output range of the process image.

The DALI Multi-Master module can be operated in "Easy" mode (default) and "Full" mode. "Easy" mode is used to transmit simply binary signals for lighting control. Configuration or programming via DALI master module is unnecessary in "Easy" mode.

Changes to individual bits of the process image are converted directly into DALI commands for a pre-configured DALI network. 22 bytes of the 24-byte process image can be used directly for switching of electronic ballasts (ECG), groups or scenes in "Easy" mode. Switching commands are transmitted via DALI and group addresses, where each DALI and each group address is represented by a 2-bit pair.

The structure of the process data is described in detail in the following tables.

Table 257: Overview of Input Process Image in the "Easy" Mode

Input process image				
Offset	Byte designation		Note	
	High byte	Low byte		
0	-	S	res.	Status, activate broadcast Bit 0: 1-/2-button mode Bit 2: Broadcast status ON/OFF Bit 1,3-7: -
1	DA4...DA7	DA0...DA3	Bitpaar für DALI-Adresse DA0: Bit 1: Bit set = ON Bit not set = OFF Bit 2: Bit set = Error Bit not set = No error Bit pairs DA1 ... DA63 similar to DA0.	
2	DA12...DA15	DA8...DA11		
3	DA20...DA23	DA16...DA19		
4	DA28...DA31	DA24...DA27		
5	DA36...DA39	DA32...DA35		
6	DA44...DA47	DA40...DA43		
7	DA52...DA55	DA48...DA51		
8	DA60...DA63	DA56...DA59		
9	GA4...GA7	GA0...GA3	Bit pair for DALI group address GA0: Bit 1: Bit set = ON Bit not set = OFF Bit 2: Bit set = Error Bit not set = No error Bit pairs GA1 ... GA15 similar to GA0.	
10	GA12...GA15	GA8...GA11		
11	-	-	Not in use	

DA = DALI address
GA = Group address

Table 258: Overview of the Output Process Image in the "Easy" Mode

Output process image				
Offset	Byte designation		Note	
	High byte	Low byte		
0	-	S	res.	Broadcast ON/OFF and activate: Bit 0: Broadcast ON Bit 1: Broadcast OFF Bit 2: Broadcast ON/OFF/dimming Bit 3: Broadcast short ON/OFF Bits 4 ... 7: reserved
1	DA4...DA7	DA0...DA3	Bit pair for DALI address DA0: Bit 1: short: DA switch ON long: dimming, brighter Bit 2: short: DA switch OFF long: dimming, darker Bit pairs DA1 ... DA63 similar to DA0.	
2	DA12...DA15	DA8...DA11		
3	DA20...DA23	DA16...DA19		
4	DA28...DA31	DA24...DA27		
5	DA36...DA39	DA32...DA35		
6	DA44...DA47	DA40...DA43		
7	DA52...DA55	DA48...DA51		
8	DA60...DA63	DA56...DA59		
9	GA4...GA7	GA0...GA3	Bitpaar für DALI-Gruppenadresse GA0: Bit 1: short: GA switch ON long: dimming, brighter Bit 2: short: GA switch OFF long: dimming, darker Bit pairs GA1 ... GA15 similar to GA0.	
10	GA12...GA15	GA8...GA11		
11	Bit 8...15	Bit 0...7	Switch scene 0...15	

DA = DALI address
GA = Group address

13.2.5.13 LON[®] FTT Module

753-648

The process image of the LON[®] FTT module consists of a control/status byte and 23 bytes of bidirectional communication data that is processed by the WAGO-I/O-PRO function block "LON_01.lib". This function block is essential for the function of the LON[®] FTT module and provides a user interface on the control side.

13.2.5.14 EnOcean Radio Receiver

750-642

The EnOcean radio receiver has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following tables illustrate the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

Table 259: EnOcean Radio Receiver 750-642

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	S	Data byte	Status byte
1	D2	D1	Data bytes	

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C	not used	Control byte
1	-	-	not used	

13.2.5.15 MP Bus Master Module

750-643

The MP Bus Master Module has a total of 8 bytes of user data in both the Input and Output Process Image (6 bytes of module data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Table 260: MP Bus Master Module 750-643

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	C1/S1	C0/S0	extended Control/Status byte	Control/status byte
1	D1	D0	Data bytes	
2	D3	D2		
3	D5	D4		

13.2.5.16 Bluetooth® RF-Transceiver

750-644

The size of the process image for the *Bluetooth*® module can be adjusted to 12, 24 or 48 bytes.

It consists of a control byte (input) or status byte (output); an empty byte; an overlay able mailbox with a size of 6, 12 or 18 bytes (mode 2); and the *Bluetooth*® process data with a size of 4 to 46 bytes.

Thus, each *Bluetooth*® module uses between 12 and 48 bytes in the process image. The sizes of the input and output process images are always the same.

The first byte contains the control/status byte; the second contains an empty byte.

Process data attach to this directly when the mailbox is hidden. When the mailbox is visible, the first 6, 12 or 18 bytes of process data are overlaid by the mailbox data, depending on their size. Bytes in the area behind the optionally visible mailbox contain basic process data. The internal structure of the *Bluetooth*® process data can be found in the documentation for the *Bluetooth*® 750-644 RF Transceiver.

The mailbox and the process image sizes are set with the startup tool WAGO-I/O-CHECK.

Table 261: Bluetooth® RF-Transceiver 750-644

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C0/S0	not used	Control/status byte
1	D1	D0	Mailbox (0, 3, 6 or 9 words) and Process data (2-23 words)	
2	D3	D2		
3	D5	D4		
...	...	...		
max. 23	D45	D44		

13.2.5.17 Vibration Velocity/Bearing Condition Monitoring VIB I/O

750-645

The Vibration Velocity/Bearing Condition Monitoring VIB I/O has a total of 12 bytes of user data in both the Input and Output Process Image (8 bytes of module data and 4 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 8 words mapped into each image. Word alignment is applied.

Table 262: Vibration Velocity/Bearing Condition Monitoring VIB I/O 750-645

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C0/S0	not used	Control/status byte (log. Channel 1, Sensor input 1)
1	D1	D0	Data bytes (log. Channel 1, Sensor input 1)	
2	-	C1/S1	not used	Control/status byte (log. Channel 2, Sensor input 2)
3	D3	D2	Data bytes (log. Channel 2, Sensor input 2)	
4	-	C2/S2	not used	Control/status byte (log. Channel 3, Sensor input 1)
5	D5	D4	Data bytes (log. Channel 3, Sensor input 3)	
6	-	C3/S3	not used	Control/status byte (log. Channel 4, Sensor input 2)
7	D7	D6	Data bytes (log. Channel 4, Sensor input 2)	

13.2.5.18 KNX/EIB/TP1 Module

753-646

The KNX/TP1 module appears in router and device mode with a total of 24-byte user data within the input and output area of the process image, 20 data bytes and 2 control/status bytes. Even though the additional bytes S1 or C1 are transferred as data bytes, they are used as extended status and control bytes. The opcode is used for the read/write command of data and the triggering of specific functions of the KNX/EIB/TP1 module. Word-alignment is used to assign 12 words in the process image. Access to the process image is not possible in router mode. Telegrams can only be tunneled.

In device mode, access to the KNX data can only be performed via special function blocks of the IEC application. Configuration using the ETS engineering tool software is required for KNX.

Table 263: KNX/EIB/TP1 Module 753-646

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	S0	not used	Status byte
1	S1	OP	extended Status byte	Opcode
2	D1	D0	Data byte 1	Data byte 0
3	D3	D2	Data byte 3	Data byte 2
4	D5	D4	Data byte 5	Data byte 4
5	D7	D6	Data byte 7	Data byte 6
6	D9	D8	Data byte 9	Data byte 8
7	D11	D10	Data byte 11	Data byte 10
8	D13	D12	Data byte 13	Data byte 12
9	D15	D14	Data byte 15	Data byte 14
10	D17	D16	Data byte 17	Data byte 16
11	D19	D18	Data byte 19	Data byte 18

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C0	not used	Control byte
1	C1	OP	extended Control byte	Opcode
2	D1	D0	Data byte 1	Data byte 0
3	D3	D2	Data byte 3	Data byte 2
4	D5	D4	Data byte 5	Data byte 4
5	D7	D6	Data byte 7	Data byte 6
6	D9	D8	Data byte 9	Data byte 8
7	D11	D10	Data byte 11	Data byte 10
8	D13	D12	Data byte 13	Data byte 12
9	D15	D14	Data byte 15	Data byte 14
10	D17	D16	Data byte 17	Data byte 16
11	D19	D18	Data byte 19	Data byte 18

13.2.5.19 AS-interface Master Module

750-655

The length of the process image of the AS-interface master module can be set to fixed sizes of 12, 20, 24, 32, 40 or 48 bytes.

It consists of a control or status byte, a mailbox with a size of 0, 6, 10, 12 or 18 bytes and the AS-interface process data, which can range from 0 to 32 bytes.

The AS-interface master module has a total of 6 to maximally 24 words data in both the Input and Output Process Image. Word alignment is applied.

The first Input and output word, which is assigned to an AS-interface master module, contains the status / control byte and one empty byte.

Subsequently the mailbox data are mapped, when the mailbox is permanently superimposed (Mode 1).

In the operating mode with suppressible mailbox (Mode 2), the mailbox and the cyclical process data are mapped next.

The following words contain the remaining process data.

The mailbox and the process image sizes are set with the startup tool WAGO-I/O-CHECK.

Table 264: AS-interface Master Module 750-655

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C0/S0	not used	Control/status byte
1	D1	D0	Mailbox (0, 3, 5, 6 or 9 words)/ Process data (0-16 words)	
2	D3	D2		
3	D5	D4		
...	...	...		
max. 23	D45	D44		

13.2.6 System Modules

13.2.6.1 System Modules with Diagnostics

750-610, -611

The modules provide 2 bits of diagnostics in the Input Process Image for monitoring of the internal power supply.

Table 265: System Modules with Diagnostics 750-610, -611

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 2 Fuse	Diagnostic bit S 1 Fuse

13.2.6.2 Binary Space Module

750-622

The Binary Space Modules behave alternatively like 2 channel digital input modules or output modules and seize depending upon the selected settings 1, 2, 3 or 4 bits per channel. According to this, 2, 4, 6 or 8 bits are occupied then either in the process input or the process output image.

Table 266: Binary Space Module 750-622 (with Behavior Like 2 Channel Digital Input)

Input and Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(Data bit DI 8)	(Data bit DI 7)	(Data bit DI 6)	(Data bit DI 5)	(Data bit DI 4)	(Data bit DI 3)	Data bit DI 2	Data bit DI 1

14 Application Examples

14.1 Test of MODBUS protocol and fieldbus nodes

You require a MODBUS master to test the function of your fieldbus node. For this purpose, various manufacturers offer a range of PC applications that you can, in part, download from the Internet as free of charge demo versions.

One of the programs which is particularly suitable to test your ETHERNET TCP/IP fieldbus node, is for instance **ModScan** from Win-Tech.

Information



Additional Information

A free of charge demo version from ModScan32 and further utilities from Win-Tech can be found in the Internet under:

<http://www.win-tech.com/html/demos.htm>

ModScan32 is a Windows application that works as a MODBUS master.

This program allows you to access the data points of your connected ETHERNET TCP/IP fieldbus node and to proceed with the desired changes.

Information



Additional Information

For a description example relating to the software operation, refer to:

<http://www.win-tech.com/html/modscan32.htm>

14.2 Visualization and Control using SCADA Software

This chapter is intended to give insight into how the WAGO ETHERNET fieldbus coupler/controller can be used for process visualization and control using standard user software.

There is a wide range of process visualization programs, called SCADA Software, from various manufacturers.

SCADA is the abbreviation for Supervisory Control and Data Acquisition.

It is a user-orientated tool used as a production information system in the areas of automation technology, process control and production monitoring.

The use of SCADA systems includes the areas of visualization and monitoring, data access, trend recording, event and alarm processing, process analysis and targeted intervention in a process (control).

The WAGO ETHERNET fieldbus node provides the required process input and output values.



Note

SCADA software has to provide a MODBUS device driver and support MODBUS/TCP functions!

When choosing suitable SCADA software, ensure that it provides a MODBUS device driver and supports the MODBUS/TCP functions in the coupler.

Visualization programs with MODBUS device drivers are available from i.e. Wonderware, National Instruments, Think&Do or KEPware Inc., some of which are available on the Internet as demo versions.

The operation of these programs is very specific. However, a few essential steps are described to illustrate the way an application can be developed using a WAGO ETHERNET fieldbus node and SCADA software in principle:

1. Load the MODBUS ETHERNET driver and select MODBUS ETHERNET
2. Enter the IP address for addressing the fieldbus node

At this point, some programs allow the user to give the node an alias name, i.e. to call the node “Measuring data”. The node can then be addressed with this name.

3. Create a graphic object, such as a switch (digital) or a potentiometer (analog)

This object is displayed on the work area.

4. Link the object to the desired data point on the node by entering the following data:
 - Node address (IP address or alias name)
 - The desired MODBUS function codes (register/bit read/write)
 - The MODBUS address of the selected channel

Entry is program specific.

Depending on the user software the MODBUS addressing of a bus module can be represented with up to 5 digits.

Example of the MODBUS Addressing

In the case of SCADA Software Lookout from National Instruments the MODBUS function codes are used with a 6 digit coding, whereby the first digit represents the MODBUS table (0, 1, 3 or 4) and implicit the function code (see following table):

Table 267: MODBUS Table and Function Codes

MODBUS table	MODBUS function code	
0	FC1 or FC15	Reading of input bits or writing of several output bits
1	FC2	Reading of several input bits
3	FC4 or FC 16	Reading of several input registers or writing of several output registers
4	FC3	Reading of several input registers

The following five digits specify the channel number (beginning with 1) of the consecutively numbered digital or analog input and/or output channels.

Examples:

- Reading/writing the first digital input: i.e. 0 0000 1
- Reading/writing the second analog input: i.e. 3 0000 2

Application Example:

Thus, the digital input channel 2 of the above node “Measuring data” can be read out with the input: “Measuring data. 0 0000 2”.

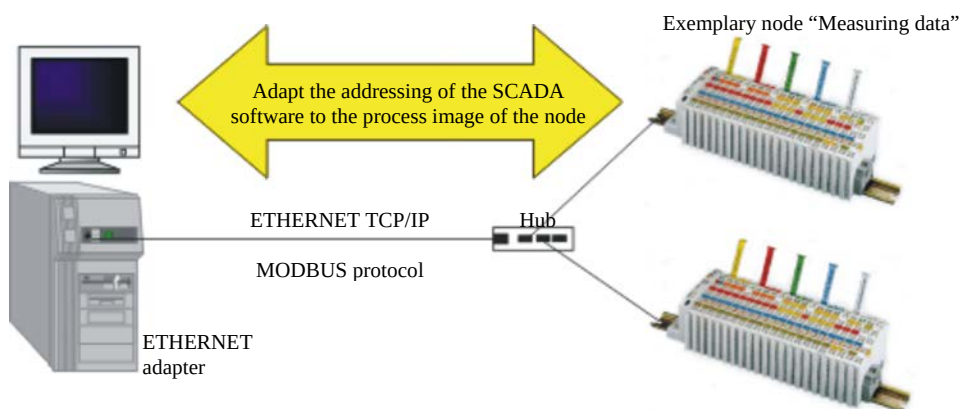


Figure 105: Example SCADA Software with MODBUS Driver



Information

Additional Information

Please refer to the respective SCADA product manual for a detailed description of the particular software operation.

15 Use in Hazardous Environments

The **WAGO-I/O-SYSTEM 750** (electrical equipment) is designed for use in Zone 2 hazardous areas.

The following sections include both the general identification of components (devices) and the installation regulations to be observed. The individual subsections of the “Installation Regulations” section must be taken into account if the I/O module has the required approval or is subject to the range of application of the ATEX directive.

15.1 Marking Configuration Examples

15.1.1 Marking for Europe According to ATEX and IEC-Ex

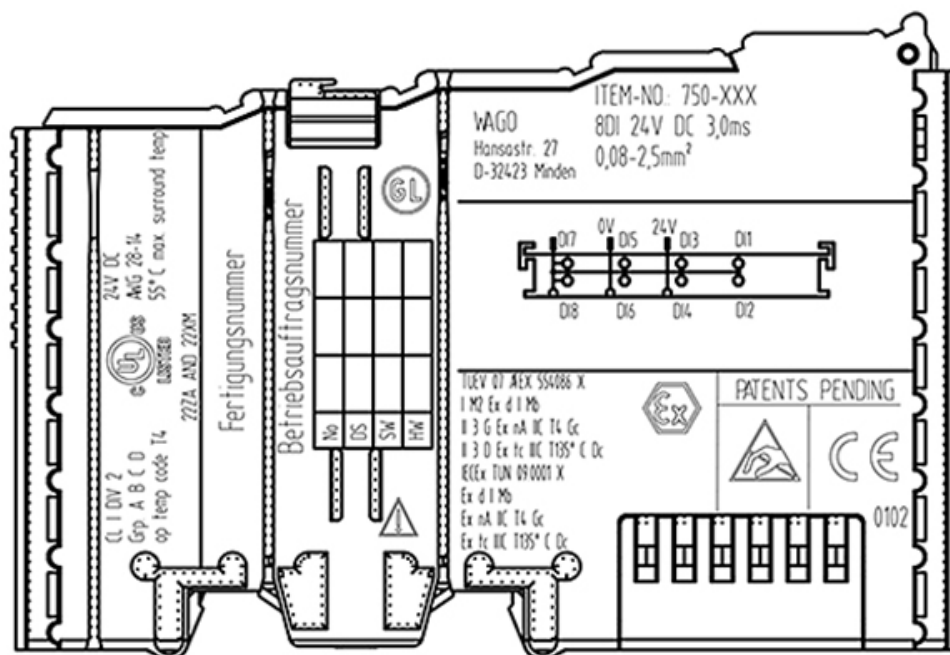


Figure 106: Side Marking Example for Approved I/O Modules According to ATEX and IECEx

TUEV 07 ATEX 554086 X
 I M2 Ex d I Mb
 II 3 G Ex nA IIC T4 Gc
 II 3 D Ex tc IIC T135° C Dc
 IECEx TUN 09.0001 X
 Ex d I Mb
 Ex nA IIC T4 Gc
 Ex tc IIC T135° C Dc



Figure 107: Text Detail – Marking Example for Approved I/O Modules According to ATEX and IECEx.

Table 268: Description of Marking Example for Approved I/O Modules According to ATEX and IECEx

Marking	Description
TÜV 07 ATEX 554086 X IECEx TUN 09.0001 X	Approving authority and certificate numbers
Dust	
II	Equipment group: All except mining
3D	Category 3 (Zone 22)
Ex	Explosion protection mark
tc Dc	Type of protection and equipment protection level (EPL): protection by enclosure
IIIC	Explosion group of dust
T 135°C	Max. surface temperature of the enclosure (without a dust layer)
Mining	
I	Equipment group: Mining
M2	Category: High level of protection
Ex	Explosion protection mark
d Mb	Type of protection and equipment protection level (EPL): Flameproof enclosure
I	Explosion group for electrical equipment for mines susceptible to firedamp
Gases	
II	Equipment group: All except mining
3G	Category 3 (Zone 2)
Ex	Explosion protection mark
nA Gc	Type of protection and equipment protection level (EPL): Non-sparking equipment
nC Gc	Type of protection and equipment protection level (EPL): Sparking apparatus with protected contacts. A device which is so constructed that the external atmosphere cannot gain access to the interior
IIC	Explosion group of gas and vapours
T4	Temperature class: Max. surface temperature 135°C

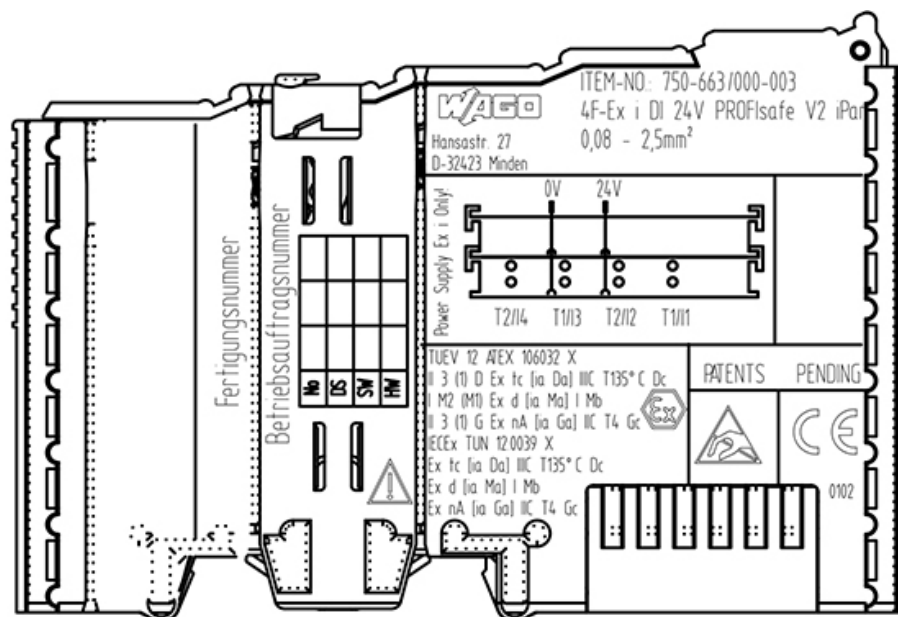


Figure 108: Side Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx.

TUEV 12 ATEX 106032 X
 II 3 (1) D Ex tc [ia Da] IIC T135° C Dc
 I M2 (M1) Ex d [ia Ma] I Mb
 II 3 (1) G Ex nA [ia Ga] IIC T4 Gc 
 IECEx TUN 12.0039 X
 Ex tc [ia Da] IIC T135° C Dc
 Ex d [ia Ma] I Mb
 Ex nA [ia Ga] IIC T4 Gc

Figure 109: Text Detail – Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx.

Table 269: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx

Marking	Description
TÜV 07 ATEX 554086 X IECE _x TUN 09.0001X	Approving authority and certificate numbers
TÜV 12 ATEX 106032 X IECE _x TUN 12.0039 X	
Dust	
II	Equipment group: All except mining
3(1)D	Category 3 (Zone 22) equipment containing a safety device for a category 1 (Zone 20) equipment
3(2)D	Category 3 (Zone 22) equipment containing a safety device for a category 2 (Zone 21) equipment
Ex	Explosion protection mark
tc Dc	Type of protection and equipment protection level (EPL): protection by enclosure
[ia Da]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 20
[ib Db]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 21
IIIC	Explosion group of dust
T 135°C	Max. surface temperature of the enclosure (without a dust layer)
Mining	
I	Equipment Group: Mining
M2 (M1)	Category: High level of protection with electrical circuits which present a very high level of protection
Ex d Mb	Explosion protection mark with Type of protection and equipment protection level (EPL): Flameproof enclosure
[ia Ma]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety electrical circuits
I	Explosion group for electrical equipment for mines susceptible to firedamp

Table 269: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx

Gases	
II	Equipment group: All except mining
3(1)G	Category 3 (Zone 2) equipment containing a safety device for a category 1 (Zone 0) equipment
3(2)G	Category 3 (Zone 2) equipment containing a safety device for a category 2 (Zone 1) equipment
Ex	Explosion protection mark
nA Gc	Type of protection and equipment protection level (EPL): Non-sparking equipment
[ia Ga]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 0
[ia Gb]	Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 1
IIC	Explosion group of gas and vapours
T4	Temperature class: Max. surface temperature 135°C

15.1.2 Marking for America According to NEC 500

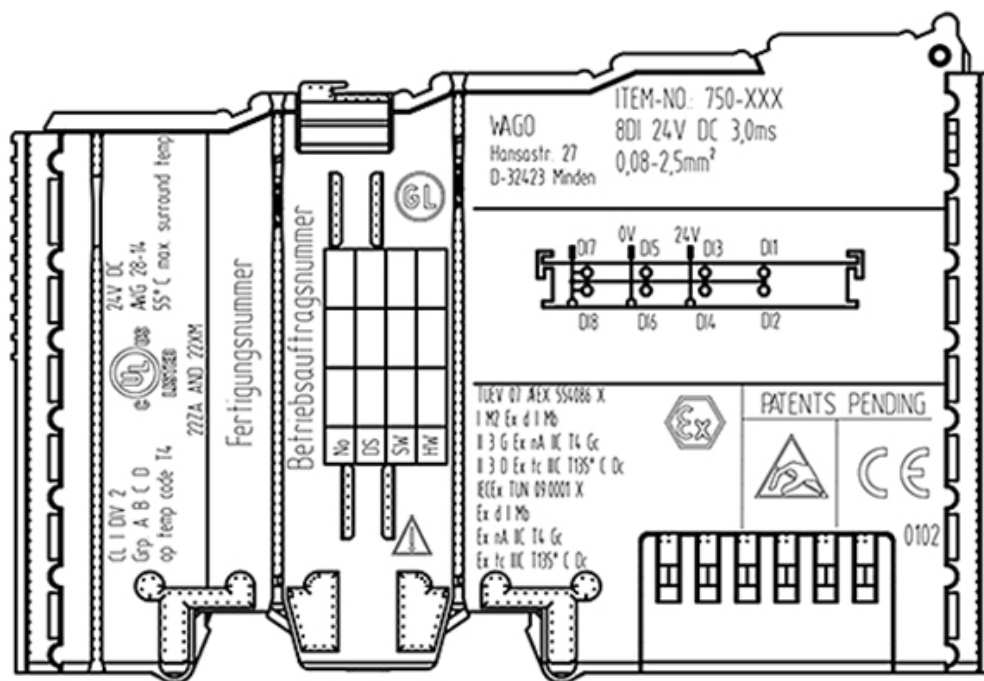


Figure 110: Side Marking Example for I/O Modules According to NEC 500

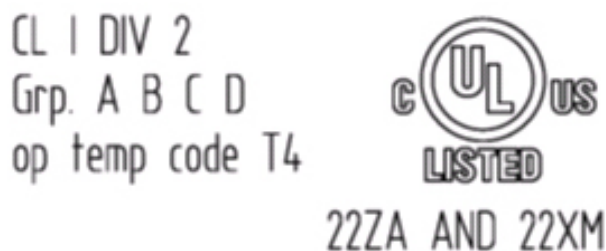


Figure 111: Text Detail – Marking Example for Approved I/O Modules According to NEC 500

Table 270: Description of Marking Example for Approved I/O Modules According to NEC 500

Marking	Description
CL I	Explosion protection group (condition of use category)
DIV 2	Area of application
Grp. ABCD	Explosion group (gas group)
Op temp code T4	Temperature class

15.2 Installation Regulations

For the installation and operation of electrical equipment in hazardous areas, the valid national and international rules and regulations which are applicable at the installation location must be carefully followed.

15.2.1 Special Conditions for Safe Use (ATEX Certificate TÜV 07 ATEX 554086 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15 and EN 60079-31.
For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64.
The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. Dip-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes.
The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
This is although and in particular valid for the interfaces “Memory-Card”, “USB”, “Fieldbus connection”, “Configuration and programming interface”, “antenna socket”, “D-Sub”, “DVI-port” and the “Ethernet interface”. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.
6. For replaceable fuses the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The following warnings shall be placed nearby the unit:
WARNING – DO NOT REMOVE OR REPLACE FUSE WHEN ENERGIZED
WARNING – DO NOT SEPARATE WHEN ENERGIZED
WARNING – SEPARATE ONLY IN A NON-HAZARDOUS AREA

15.2.2 Special Conditions for Safe Use (ATEX Certificate TÜV 12 ATEX 106032 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** Ex i shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15 and EN 60079-31. For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
4. For the type the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II (non mains/mains circuits) as defined in EN 60664-1.

15.2.3 Special Conditions for Safe Use (IEC-Ex Certificate TUN 09.0001 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15 and IEC 60079-31. For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64.
The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
This is although and in particular valid for the interfaces “Memory-Card”, “USB”, “Fieldbus connection”, “Configuration and programming interface”, “antenna socket”, “D-Sub”, “DVI-port” and the “Ethernet interface”. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.
6. For replaceable fuses the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The following warnings shall be placed nearby the unit:
WARNING – DO NOT REMOVE OR REPLACE FUSE WHEN ENERGIZED
WARNING – DO NOT SEPARATE WHEN ENERGIZED
WARNING – SEPARATE ONLY IN A NON-HAZARDOUS AREA

15.2.4 Special Conditions for Safe Use (IEC-Ex Certificate IECEx TUN 12.0039 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus independent I/O Modules WAGO-I/O-SYSTEM 750-*** Ex i shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15, IEC 60079-31.
For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64.
The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes.
The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
4. For the type the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II (non mains/mains circuits) as defined in IEC 60664-1.

15.2.5 Special Conditions for Safe Use According to ANSI/ISA 12.12.01

- A. “This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only.”
- B. “This equipment is to be fitted within tool-secured enclosures only.”
- C. “WARNING Explosion hazard - substitution of components may impair suitability for Class I, Div. 2.”
- D. “WARNING – Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous” has to be placed near each operator accessible connector and fuse holder.
- E. When a fuse is provided, the following information shall be provided: “A switch suitable for the location where the equipment is installed shall be provided to remove the power from the fuse.”
- F. For devices with EtherCAT/Ethernet connectors “Only for use in LAN, not for connection to telecommunication circuits.”
- G. “WARNING - Use Module 750-642 only with antenna module 758-910.”
- H. For Couplers/Controllers and Economy bus modules only: The instructions shall contain the following: “The configuration interface Service connector is for temporary connection only. Do not connect or disconnect unless the area is known to be non-hazardous. Connection or disconnection in an explosive atmosphere could result in an explosion.”
- I. Modules containing fuses only: “WARNING - Devices containing fuses must not be fitted into circuits subject to over loads, e.g. motor circuits.”
- J. Modules containing SD card reader sockets only: “WARNING - Do not connect or disconnect SD-Card while circuit is live unless the area is known to be free of ignitable concentrations of flammable gases or vapors.”



Information

Additional Information

Proof of certification is available on request.

Also take note of the information given on the operating and assembly instructions.

The manual, containing these special conditions for safe use, must be readily available to the user.

16 Appendix

16.1 List of all BACnet Properties in Native Operation

This section contains a list of all properties of BACnet Objects that can be used in native operation.

The properties of BACnet Objects are classified as follows:

- Individual, primitive data types
- Complex data types consisting of several data types
- Collection of data elements of the same type

In some cases, the number of data elements of a collection is specified; sometimes the number of elements can vary. In some cases, the elements must be accessed individually or there is a certain sequence to follow. BACnet provides two forms of data types for properties that represent a collection of data elements of one type:

- „BACnetARRAY”
- „List of”

The data type "**BACnetARRAY**" describes a structured field and consists of data elements of a data type in an ordered sequence. A "field index", i.e. an unsigned integer value, can be used to access the individual components of a field property (write or read access). An index of 0 (zero) indicates that the number of present data elements has been returned. If there is no field index, this means that all elements of the field can be accessed. A field index of n greater than zero indicates the nth element of the sequence. When using field properties with BACnet objects, the designation "BACnetARRAY[N] of data type" indicates an ordered sequence of data elements with this data type. If the size of a field can be changed by writing in the field, the field element 0 must be writable. If the value of the field element 0 is reduced, this is shortened and all elements of the field with an index greater than the new value of the field element 0 are deleted. If the value of the field element 0 increases, new elements of the field with an index greater than the old value of the field element 0 are replaced. Values that are assigned to these elements are local values if not otherwise indicated. If the size of a field can be changed, writing in the entire field as a single property with a different number of elements leads to a change in field size. The attempt to write with an index greater than the field in a field element leads to an error, and the field is not enlarged in order to accept the element. Fields with a size established in the Standard cannot be enlarged or reduced in size.

The data type "**List of**" is a structure list with a sequence of zero or more data elements of one data type. The length of each "List of" is different. If not required for a certain purpose, no maximum size should be given for a "List of". The designation "List of <data type>" indicates a sequence of zero or more data elements of the corresponding type.

The difference between the properties "BACnetARRAY" and "List of" is that the elements of a field can be individually accessed through a field index while this is not possible for the elements of a "List of". Furthermore, the number of elements of a "BACnetARRAY" can be determined by reading the field index 0; the number of elements in a "List of" can only be determined by reading the whole value of the property and conducting a count.

In the following list, the associated BACnet data type is indicated for each property. Under "Object", all objects in which the respective property occurs are listed. The description explains functions and interrelationships of the properties in question. If the meanings of the properties are different for different objects, the explanations for these objects are given separately.

16.1.1 Acked_Transitions

Data Type	BACnetEventTransitionBits
Object	Binary Input, Binary Output, Analog Input, Analog Output
Description	This property of the type BACnetEventTransitionBits transmits three flags that indicate the receipt of the confirmation for TO-OFFNORMAL, TO-FAULT and TO-NORMAL events. In the context of the Analog Input objects, transitions the High_Limit and Low_Limit Event_States are considered to be OFFNORMAL events. The flags are deleted after the corresponding event has taken place and set under the following conditions: • upon receipt of the corresponding acknowledgment; • upon the occurrence of the event if the corresponding flag is not set in the Event_Enable property (meaning event notifications will not be generated for this condition and thus no acknowledgment is expected); • upon the occurrence of the event if the corresponding flag is set in the Event_Enable property and the corresponding flag in the Ack_Required property of the Notification Class object implicitly referenced by the Notification_Class property of this object is not set (meaning no acknowledgment is expected).

This property is required if intrinsic reporting is supported by this object.

16.1.2 Active_COV_Subscriptions

Data Type	List of BACnetCOVSubscription
Object	Device
Description	The Active_COV_Subscriptions property is a List of BACnetCOVSubscription, each of which consists of a Recipient, a Monitored Property Reference, an Issue Confirmed Notifications flag, a Time Remaining value and an optional COV Increment. This property provides a network-visible indication of those COV subscriptions that are active at any given time. Whenever a COV Subscription is created with the SubscribeCOV or SubscribeCOVProperty service, a new entry is added to the Active_COV_Subscriptions list. Similarly, whenever a COV Subscription is terminated, the corresponding entry is removed from the Active_COV_Subscriptions list. This property is required if the device supports execution of either SubscribeCOV or SubscribeCOVProperty service.

16.1.3 Active_Text

Data Type	CharacterString
Object	Binary input, Binary output
Description	This property, of type CharacterString, characterizes the intended effect of the ACTIVE state of the Present_Value property from the human operator's viewpoint. The content of this string is determined locally, but it is intended to contain a description of the ACTIVE state that can be read by the user.

Example for Binary Input:

For example, if the physical input is a switch contact, then the Active_Text property might be assigned a value such as "Fan 1 On". If either the Active_Text property or the Inactive_Text property is present, then both of them shall be present.

Example for Binary Output:

For example, if the physical output that ultimately controls the device from the operator's point of view is a relay contact that turns on a light, then the Active_Text property might be assigned a value such as "Light On". If one of the optional properties Inactive_Text or Active_Text is present, then both of these properties should be present.

16.1.4 Alarm_Value

Data Type	BACnetBinaryPV
Object	Binary input
Description	This property, of the type BACnetBinaryPV, specifies the value that the Present_Value property must have before an event is generated. This property is required if intrinsic reporting is supported by this object.

Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

- the Present_Value must maintain the value specified by Alarm_Value for a minimum period of time, specified in the Time_Delay property, and
- the TO-OFFNORMAL flag must be enabled in the Event_Enable property.

Conditions for Generating a TO-NORMAL Event

Once equal, the Present_Value must become not equal to this property before a TO-NORMAL event is generated under these conditions:

- The Present_Value must remain not equal to the Alarm_Value for a minimum period of time, specified by the Time_Delay property, and
- The TO-NORMAL flag must be enabled in the Event_Enable property.

16.1.5 APDU_Segment_Timeout

Data Type	Unsigned
Object	Device

Description	The APDU_Segment_Timeout property, of type Unsigned, shall indicate the amount of time in milliseconds between retransmission of an APDU segment. The default value for this property shall be 2000 milliseconds. This value cannot be zero if the device object property called Number_Of_APDU_Retries is not zero. If segmentation of any kind is supported, then the APDU_Segment_Timeout property must be present. In order to achieve reliable communication, it is recommended that the values of the APDU_Segment_Timeout properties of the Device objects of all intercommunicating devices should contain the same value.

16.1.6 APDU_Timeout

Data Type	Unsigned
Objects	Device
Description	The APDU_Timeout property, of type Unsigned, shall indicate the amount of time in milliseconds between retransmissions of an APDU requiring acknowledgment for which no acknowledgment has been received. The default value for this property is 3000 milliseconds for devices that permit modification of this parameter. Otherwise, the default value is 60,000 milliseconds. This value cannot be zero if the device object property called Number_Of_APDU_Retries is not zero. In order to achieve reliable communication, it is recommended that the values of the APDU_Timeout properties of the Device objects of all intercommunicating devices should contain the same value.

16.1.7 Application_Software_Version

Data Type	CharacterString
Objects	Device
Description	This property, of type CharacterString, identifies the version of application software installed in the machine. The content of this string is a local matter, but it could be a date-and-time stamp, a programmer's name, a host file version number, etc.

16.1.8 Archive

Data Type	BOOLEAN
Objects	File
Description	This property, of type BOOLEAN, indicates whether the File object has been saved for historical or backup purposes. This property shall be logical TRUE only if no changes have been made to the file data by internal processes or through File Access Services since the last time the object was archived.

16.1.9 Backup_Failure_Timeout

Data Type	Unsigned16
Objects	Device
Description	This property, of the type Unsigned16, is the time, in seconds, that the device being backed up or restored must wait before unilaterally ending the backup or restore procedure. This property must be writable with the intent that the device performing the backup, or the human operator, will configure this with an appropriate timeout.

16.1.10 Change_Of_State_Count

Data Type	Unsigned
Objects	Binary Input, Binary Output

Description This property, of the type Unsigned, represents the number of times that the Present_Value property has changed state since the Change_Of_State_Count property was most recently set to a zero value. The Change_Of_State_Count property has a range of from 0 to 65535 or greater. A "change of state" is defined as any event that alters the Present_Value property.

Binary Input:

When Out_Of_Service is FALSE, a change to the Polarity property changes the Present_Value property and is therefore considered a change of state. When Out_Of_Service is TRUE, changes to Polarity do not cause changes of state. If one of the optional properties Change_Of_State_Time, Change_Of_State_Count, or Time_Of_State_Count_Reset is present, then all of these properties must be present.

Binary Output:

A change in the Polarity property is not regarded as a change of state. If one of the optional properties Change_Of_State_Time, Change_Of_State_Count, or Time_Of_State_Count_Reset is present, then all of these properties must be present.

16.1.11 Change_Of_State_Time

Data Type BACnetDateTime

Objects Binary Input, Binary Output

Description This property, of the type BACnetDateTime, represents the date and time at which the most recent change of state occurred. A "change of state" is defined as any event that alters the Present_Value property.

Binary Input:

When Out_Of_Service is FALSE, a change to the Polarity property changes the Present_Value property is therefore considered a change of state. When Out_Of_Service is TRUE, changes to Polarity shall not cause changes of state. If one of the optional properties Change_Of_State_Time, Change_Of_State_Count, or Time_Of_State_Count_Reset is present, then all of these properties must be present.

Binary Output:

A change in polarity does not lead to a change in state. If one of the optional properties Change_Of_State_Time, Change_Of_State_Count, or Time_Of_State_Count_Reset is present, then all of these properties must be present.

16.1.12 Configuration_Files

Data Type	BACnetARRAY[N] for BACnetObjectIdentifier
Objects	Device
Description	This property is a BACnet field for BACnetObjectIdentifier. Entries in the array identify the files within the device that define the device's image that can be backed up. The content of this property is only required to be valid during the backup procedure.

16.1.13 COV_Increment

Data Type	REAL
Objects	Analog input, Analog output
Description	This property of the type REAL specifies the minimum change in the property Present_Value that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

16.1.14 Database_Revision

Data Type	Unsigned
Objects	Device
Description	This property, of type Unsigned, is a logical revision number for the device's database. It is incremented when an object is created, an object is deleted, an object's name is changed, an object's Object_Identifier property is changed, or a restore is performed.

16.1.15 Data_List

Data Type	List of BACnetCalendarEntry
Objects	Calendar

Description	This property is a List of BACnetCalendarEntry, each of which is either an individual date (Date), range of dates (BACnetDateRange), or month/week-of-month/day-of-week specification (BACnetWeekNDay). If the current date fulfills the criteria of the calendar entry, the current value of the Calendar Object is TRUE. Individual fields of the various constructs may also be unspecified in which case the field acts as a "wildcard" for determining if the current date results in a match. In a date range, for example, if the startDate is unspecified, it means "any date up to and including the endDate". If the endDate is unspecified, it means "any date from the startDate on. If the calendar entry were a BACnetWeekNDay with unspecified month and week-of-month fields but with a specific day-of-week, it would mean the Calendar object would be TRUE on that day-of-week all year long. If a BACnet Device permits writing to the Date_List property, all choices in the BACnetCalendarEntry shall be permitted.

16.1.16 Daylight_Savings_Status

Data Type	BOOLEAN
Objects	Device
Description	The Daylight_Savings_Status property, of type BOOLEAN, shall indicate whether daylight savings time is in effect (TRUE) or not (FALSE) at the BACnet Device's location.

16.1.17 Deadband

Data Type	REAL
Objects	Analog input, Analog output

Description	Description: This property, of type REAL, shall specify a range between the High_Limit and Low_Limit properties, which the Present_Value property must remain within for a TO-NORMAL event to be generated under these conditions: • the Present_Value must fall below the High_Limit minus Deadband, and • the Present_Value must exceed the Low_Limit plus the Deadband, and • the Present_Value must remain within this range for a minimum period of time, specified in the Time_Delay property, and • either the HighLimitEnable or LowLimitEnable flag must be set in the Limit_Enable property, and • the TO-NORMAL flag must be enabled in the Event_Enable property. This property is required if intrinsic reporting is supported by this object.

16.1.18 Description

Data Type	CharacterString
Objects	File, Binary input, Binary output, Analog input, Analog output, Device, Calendar, Scheduler
Description	This property, of type CharacterString, is a string of printable characters that may be used to describe the application being carried out by the BACnet Device or other locally desired descriptive information.

16.1.19 Device_Address_Binding

Data Type	List of BACnetAddressBinding
Objects	Device

Description	The Device_Address_Binding property is a List of BACnetAddressBinding each of which consists of a BACnet Object_Identifier of a BACnet Device object and a BACnet device address in the form of a BACnetAddress. Entries in the list identify the actual device addresses that will be used when the remote device must be accessed via a BACnet service request. A value of zero is used for the network number portion of BACnetAddress entries for other devices residing on the same network as this device. The list may be empty if no device identifier-device address bindings are currently known to the device.

16.1.20 Device_Type

Data Type	CharacterString
Objects	Binary Input, Binary Output, Analog Input, Analog Output
Description	This property, of type CharacterString, is a text description of the physical device connected to a binary input, binary output, analog input or analog output. It will typically be used to describe the type of device attached to this input/output.

16.1.21 Effective_Period

Data Type	BACnetDateRange
Objects	Scheduler
Description	This property specifies the range of dates within which the Schedule object is active. Seasonal scheduling may be achieved by defining several Schedule Objects with non-overlapping Effective_Periods to control the same property references. Upon entering its effective period, the object calculates its Present_Value and returns this value to all members of the List_Of_Object_Property_References property. An error writing to any member of the list does not stop the Schedule Object from writing to the remaining members.

16.1.22 Elapsed_Active_Time

Data Type	Unsigned32
Objects	Binary input, Binary output

Description	This property, of type Unsigned32, represents the accumulated number of seconds that the Present_Value property has had the value ACTIVE since the Elapsed_Active_Time property was most recently set to a zero value.

16.1.23 Event_Enable

Data Type	BACnetEventTransitionBits
Objects	Binary Input, Binary Output, Analog Input, Analog Output
Description	This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be "offnormal" events.

16.1.24 Event_State

Data Type	BACnetEventState
Objects	Binary Input, Binary Output, Analog Input, Analog Output
Description	The Event_State property, of type BACnetEventState, is included in order to provide a way to determine if this object has an active event state associated with it. If the object supports intrinsic reporting, then the Event_State property indicates the event state of the object. If the object does not support intrinsic reporting, then the value of this property shall be NORMAL. If the Reliability property is present and does not have the value NO_FAULT_DETECTED, then the value of the Event_State property is FAULT. A change in the Event_State property to the value FAULT is considered a Fault Event.

16.1.25 Event_Time_Stamps

Data Type	BACnetARRAY[3] for BACnetTimeStamp
Objects	Binary Input, Binary Output, Analog Input, Analog Output

Description	This optional property, of type BACnetARRAY[3] of BACnetTimeStamp conveys the times of the last event notifications for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. Time stamps of type Time or Date must have 'FF' in each octet and Sequence number time stamps must have the value 0 if no event notification of that type has been generated since the object was created.

16.1.26 Exception_Schedule

Data Type	BACnetARRAY[N]of BACnetSpecialEvent
Objects	Scheduler
Description	This property is a BACnetARRAY of BACnetSpecialEvents. Each BACnetSpecialEvent describes a sequence of schedule actions that takes precedence over the normal day's behavior on a specific day or days. BACnetSpecialEvent ::= (Period, List of BACnetTimeValue, EventPriority) Period ::= Choice of {BACnetCalendarEntry CalendarReference} EventPriority ::= Unsigned (1..16)

The Period may be a BACnetCalendarEntry or it may refer to a Calendar object. A BACnetCalendarEntry would be used if the Exception_Schedule is specific to this Schedule object. A calendar might be defined for common holidays to be referenced by multiple Schedule Objects. Each BACnetCalendarEntry is either an individual date (Date), range of dates (BACnetDateRange), or month/week-of-month/day-of-week specification (BACnetWeekNDay). If the current date matches any of the calendar entry criteria, the Exception Schedule would be activated and the list of BACnetTimeValues would be enabled for use.

Individual fields of the various constructs of the BACnetCalendarEntry may also have a "wildcard" value used for determining if the current date falls within the Period of the Exception Schedule. In a date range, for example, if the startDate is a wildcard, it means "any date up to and including the endDate". If the endDate is a wildcard, it means "any date from the startDate on". If the calendar entry were a BACnetWeekNDay with wildcard for month and week-of-month fields but with a specific day-of-week, it would mean the Exception Schedule would apply on that day-of-week all year long.

Each BACnetSpecialEvent contains an EventPriority that determines its importance relative to other BACnetSpecialEvents within the same Exception_Schedule. Since SpecialEvents within the same Exception_Schedule may have overlapping periods, it is necessary to have a mechanism to determine the relative priorities for the SpecialEvents that apply on any given day. If more than one SpecialEvent applies to a given day, the relative priority of the SpecialEvents shall be determined by their EventPriority values. If multiple overlapping SpecialEvents have the same EventPriority value, then the SpecialEvent with the lowest index number in the array shall have higher relative priority. The highest EventPriority is 1 and the lowest is 16. The EventPriority is not related to the Priority_For_Writing property of the Schedule Object. If a BACnet Device supports writing to the Exception_Schedule property, all possible choices in the BACnetSpecialEvents shall be supported. If the size of this array is increased by writing to array index zero, each new array element shall contain an empty List of BACnetTimeValue. If the Exception_Schedule property is written with a schedule item containing a data type not supported by this instance of the Schedule object (e.g., the List_Of_Object_Property_References property cannot be configured to reference a property of the unsupported data type), the device may return a Result(-) response, specifying an 'Error Class' of PROPERTY and an 'Error Code' of DATATYPE_NOT_SUPPORTED.

16.1.27 Feedback_Value

Data Type	BACnetBinaryPV
Objects	Binary output
Description	This property, of type BACnetBinaryPV, shall indicate the status of a feedback value from which the Present_Value must differ before an event is generated.

16.1.28 File_Access_Method

Data Type	BACnetFileAccessMethod
Objects	File
Description	This property, of type BACnetFileAccessMethod, indicates the type(s) of file access supported for this object. The possible values for File_Access_Method are: {RECORD_ACCESS, STREAM_ACCESS}.

16.1.29 File_Size

Data Type	Unsigned
Objects	File
Description	This property, of type Unsigned, indicates the size of the file data in octets. If the size of the file can be changed by writing to the file, and File_Access_Method is STREAM_ACCESS, then this property shall be writable. Writing to the File_Size property with a value less than the current size of the file truncates the file at the specified position. Writing a File_Size of 0 shall delete all of the file data but not the File object itself. Writing to the File_Size property with a value greater than the current size of the file expands the size of the file, but the value of the new octets of the file are determined locally.

16.1.30 File_Type

Data Type	CharacterString
Objects	File
Description	This property, of type CharacterString, identifies the intended use of this file.

16.1.31 Firmware_Revision

Data Type	CharacterString
Objects	Device
Description	This property, of type CharacterString, is assigned by the vendor to represent the level of firmware installed in the BACnet Device.

16.1.32 High_Limit

Data Type	REAL
Objects	Analog Input, Analog Output

Description	This property, of type REAL, shall specify a limit that the Present_Value must exceed before an event is generated.

16.1.33 Inactive_Text

Data Type	CharacterString
Objects	Binary input, Binary output
Description	Binary Input: This property, of type CharacterString, characterizes the intended effect of the INACTIVE state of the Present_Value property from the human operator's viewpoint. The content of this string is determined locally, but it is intended to contain a description of the ACTIVE state that can be read by the user. For example, if the physical input is connected to a switch contact, then the Inactive_Text property might be assigned a value such as "Fan 1 Off". If either the Inactive_Text property or the Active_Text property is present, then both of them shall be present. Binary Output: This property, of the type CharacterString, characterizes the intended effect, from the human operator's viewpoint, of the INACTIVE state of the Present_Value property on the final device that is ultimately controlled by the output. The content of this character string is locally determined, but it is intended to contain description of the INACTIVE state that can be read by the user. For example, if the physical output is a relay contact that turns on a light, then the Inactive_Text property might be assigned a value such as "Light Off". If one of the optional properties Inactive_Text or Active_Text is present, then both of these properties must be present.

16.1.34 Last_Restore_Time

Data Type	BACnetTimeStamp
Objects	Device
Description	This optional property, of type BACnetTimeStamp, is the time at which the device's image was last restored.

16.1.35 Limit_Enable

Data Type	BACnetLimitEnable
Objects	Analog input, Analog output
Description	This property, of type BACnetLimitEnable, shall convey two flags that separately enable and disable reporting of high limit and low limit offnormal events and their return to normal. This property is required if intrinsic reporting is supported by this object.

16.1.36 List_Of_Object_Property_References

Data Type	List of BACnetDeviceObjectPropertyReference
Objects	Scheduler
Description	This property specifies the Device Identifiers, Object Identifiers and Property Identifiers of the properties to be written with specific values at specific times on specific days. If this property is writable, it may be restricted to only support references to objects inside of the device containing the Schedule object. If the property is restricted to referencing objects within the containing device, an attempt to write a reference to an object outside the containing device into this property shall cause a Result(-) to be returned with an error class of PROPERTY and an error code of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED. If this property is set to reference an object outside the device containing the Schedule object, the method used for writing to the referenced property value for the purpose of controlling the property is a local matter. The only restriction on the method of writing to the referenced property is that the scheduling device be capable of using WriteProperty for this purpose so as to be interoperable with all BACnet devices.

16.1.37 Local_Date

Data Type	Date
Objects	Device

Description	The Local_Date property, of type Date, shall indicate the date to the best of the device's knowledge. If the BACnet Device does not have any knowledge of time or date, then the Local_Date property shall be omitted.

16.1.38 Local_Time

Data Type	Time
Objects	Device
Description	The Local_Time property, of type Time, shall indicate the time of day to the best of the device's knowledge. If the BACnet Device does not have any knowledge of time or date, then the Local_Time property shall be omitted.

16.1.39 Location

Data Type	CharacterString
Objects	Device
Description	This property, of type CharacterString, indicates the physical location of the BACnet Device.

16.1.40 Low_Limit

Data Type	REAL
Objects	Analog input, Analog output
Description	This property, of type REAL, shall specify a limit that the Present_Value must fall below before an event is generated. This property is required if intrinsic reporting is supported by this object.

16.1.41 Max_APDU_Length_Accepted

Data Type	Unsigned
Objects	Device

Description	This property, of type Unsigned, is the maximum number of octets that may be contained in a single, indivisible application layer protocol data unit. The value of this property shall be greater than or equal to 50. The value of this property is also subject to the limitations of the data link technology used.

16.1.42 Max_Pres_Value

Data Type	REAL
Objects	Analog input, Analog output
Description	This property, of type REAL, indicates the highest number in engineering units that can be reliably obtained for the Present_Value property of this object.

16.1.43 Max_Segments_Accepted

Data Type	Unsigned
Objects	Device
Description	The Max_Segments_Accepted property, of type Unsigned, shall indicate the maximum number of segments of an APDU that this device will accept.

16.1.44 Message_Text (512)

Data Type	BACnetARRAY[3] of CharacterString
Objects	Analog Input, Analog Output, Binary Input, Binary Output, Calendar, Device, File, Scheduler
Description	The property message_text is an optional property of the character string type, with which a message text can be sent with an "event notification".

16.1.45 Min_Pres_Value

Data Type	REAL
Objects	Analog input, Analog output

Description	This property, of type REAL, indicates the lowest number in engineering units that can be reliably obtained for the Present_Value property of this object.

16.1.46 Minimum_Off_Time

Data Type	Unsigned32
Objects	Binary output
Description	This property, of type Unsigned32, represents the minimum number of seconds that the Present_Value shall remain in the INACTIVE state after a write to the Present_Value property causes that property to assume the INACTIVE state. The mechanism used for this is described in the WAGO BACnet/IP Controller 750-830 manual in the section "Prioritization".

16.1.47 Minimum_On_Time

Data Type	Unsigned32
Objects	Binary output
Description	This property, of type Unsigned32, represents the minimum number of seconds that the Present_Value shall remain in the ACTIVE state after a write to the Present_Value property causes that property to assume the ACTIVE state. The mechanism used for this is described in the WAGO BACnet/IP Controller 750-830 manual in the section "Prioritization".

16.1.48 Model_Name

Data Type	CharacterString
Objects	Device
Description	This property, of type CharacterString, is assigned by the vendor to represent the model of the BACnet Device.

16.1.49 Modification_Date

Data Type	BACnetDateTime
Objects	File
Description	This property, of type BACnetDateTime, indicates the last time this object was modified. A File object shall be considered modified when it is created or written to.

16.1.50 Notification_Class

Data Type	Unsigned
Objects	Binary Input, Binary Output, Analog Input, Analog Output
Description	This property, of type Unsigned, shall specify the notification class to be used when handling and generating event notifications for this object. The Notification_Class property implicitly refers to a Notification Class object that has a Notification_Class property with the same value.

16.1.51 Notify_Type

Data Type	BACnetNotifyType
Objects	Binary Input, Binary Output, Analog Input, Analog Output
Description	This property, of type BACnetNotifyType, shall convey whether the notifications generated by the object should be Events or Alarms.

16.1.52 Number_Of_APDU_Retries

Data Type	Unsigned
Objects	Device
Description	The Number_Of_APDU_Retries property, of type Unsigned, shall indicate the maximum number of times that an APDU shall be retransmitted. The default value for this property shall be 3. If this device does not perform retries, then this property is set to zero. If the value of this property is greater than zero, a non-zero value is placed in the device object APDU_Timeout property.

16.1.53 Object_Identifier

Data Type	BACnetObjectIdentifier
Objects	Calendar, File, Binary input, Binary output, Analog input, Analog output, Device, Scheduler
Description	This property, of the type BACnetObjectIdentifier, is a numeric code that is used to identify the object. It must be unique within the BACnet Device that maintains it. For the device object, the object identifier must be unique internetwork-wide.

16.1.54 Object_List

Data Type	BACnetARRAY[N] for BACnetObjectIdentifier
Objects	Device
Description	This read only property is a BACnetARRAY of Object_Identifiers, one Object_Identifier for each object within the device that is accessible through BACnet services.

16.1.55 Object_Name

Data Type	CharacterString
Objects	Device, Calendar, File, Binary input, Binary output, Analog input, Analog output, Scheduler
Description	This property, of the type CharacterString, represents a name for the object that is unique within the BACnet Device that maintains it. The device object is unique internetwork-wide. The minimum length of the string is one character. The set of characters used in the Object_Name is restricted to printable characters.

16.1.56 Object_Type

Data Type	BACnetObjectType
Objects	Device, Calendar, File, Binary input, Binary output, Analog input, Analog output, Scheduler

Description	This property, of the type BACnetObjectType, indicates membership in a particular object type class. The value of this property for native objects can be DEVICE, CALENDAR, FILE, BINARY_INPUT, BINARY_OUTPUT, ANALOG_INPUT, ANALOG_OUTPUT or SCHEDULE.

16.1.57 Out_Of_Service

Data Type	BOOLEAN
Objects	Binary input, Binary output, Analog input, Analog output, Scheduler
Description	Description Binary Input/Analog Input/Binary Output/Analog Output: The Out_Of_Service property, of the type BOOLEAN, indicates whether the physical input or output that the object represents is active (TRUE) or not (FALSE). If the value of Out_Of_Service is TRUE, the property Present_Value is not coupled with the physical input or output and changes are not carried out here. In addition, the Reliability property and the corresponding state of the FAULT flag of the Status_Flags property are decoupled from the physical input or output when Out_Of_Service is TRUE. While the Out_Of_Service property is TRUE, the Present_Value and Reliability properties may be changed to any value as a means of simulating specific fixed conditions or for testing purposes. Other functions that depend on the state of the Present_Value or Reliability properties must react to changes in value while Out_Of_Service is TRUE, as if those changes had occurred in the physical input or output. For the Output Objects, the property Present_Value can still be controlled via the BACnet Command Prioritization Mechanism if Out_Of_Service is TRUE. Schedule: The Out_Of_Service property, of the type BOOLEAN, indicates whether the internal calculations of the schedule object are used to determine the value of the Present_Value property (TRUE) or not (FALSE). This means that the Present_Value property is decoupled from the internal calculations and will not track changes to other properties when Out_Of_Service is TRUE. Other functions that depend on the state of the Present_Value, such as writing to the members of the List_Of_Object_Property_References, must react to changes made to that property while Out_Of_Service is TRUE, as if those changes had occurred by internal calculations.

16.1.58 Polarity

Data Type	BACnetPolarity
Objects	Binary input, Binary output
Description	This property, of type BACnetPolarity, indicates the relationship between the physical state of the input/output and the logical state represented by the Present_Value property. If the Polarity property is NORMAL, then the ACTIVE state of the Present_Value property is also the ACTIVE or ON state of the physical input/output as long as Out_Of_Service is FALSE. If the Polarity property is REVERSE, then the ACTIVE state of the Present_Value property is the INACTIVE or OFF state of the physical input/output as long as Out_Of_Service is FALSE.

Binary Input:

Therefore, when Out_Of_Service is FALSE for a constant physical input state, a change in the Polarity property shall produce a change in the Present_Value property. If Out_Of_Service is TRUE, then the Polarity property shall have no effect on the Present_Value property (see following table).

Binary Output:

If Out_Of_Service is TRUE, then the Polarity property has no effect on the physical output state (see following table).

Table 271: Interrelationships for the Polarity Property

Present_Value	Polarity	Physical state of the input or output	Physical state of device
INACTIVE	NORMAL	OFF or INACTIVE	<u>not</u> running
ACTIVE	NORMAL	ON or ACTIVE	running
INACTIVE	REVERSE	ON or ACTIVE	<u>not</u> running
ACTIVE	REVERSE	OFF or INACTIVE	running

16.1.59 Present_Value

Data Type	see Objects
Objects	Calendar (BOOLEAN), Binary input (BACnetBinaryPV), Binary output (BACnetBinaryPV), Analog input (REAL), Analog output (REAL), Scheduler (Any)

Description**Calendar:**

This property, of the type BOOLEAN, indicates the current value of the calendar: TRUE if the current date is in the Date_List and FALSE if it is not.

Binary Input:

For this object, the property Present_Value of the type BACnetBinaryPV indicates the logical state of the Binary Input. The logical state of the input is either INACTIVE or ACTIVE. The relationship between the Present_Value and the physical state of the input is determined by the Polarity property. Possible states are summarized in following Table.

Table 272: Interrelationships for the Polarity Property (Binary Input)

Present_Value	Polarity	Physical state of the input	Physical state of device
INACTIVE	NORMAL	OFF or INACTIVE	<u>Not in operation</u>
ACTIVE	NORMAL	ON or ACTIVE	In operation
INACTIVE	REVERSE	ON or ACTIVE	<u>Not in operation</u>
ACTIVE	REVERSE	OFF or INACTIVE	In operation

The Present_Value property shall be writable when Out_Of_Service is TRUE.

Binary Output:

For this object, the property of the type BACnetBinaryPV indicates the logical state of the Binary Output. The logical state of the output is either INACTIVE or ACTIVE. The relationship between the Present_Value and the physical state of the output is determined by the Polarity property. Possible states are summarized in the following table.

Table 273: Interrelationships for BACnet Polarity (Binary Output)

Present_Value	Polarity	Physical state of the output	Physical state of device
INACTIVE	NORMAL	OFF or INACTIVE	<u>Not in operation</u>
ACTIVE	NORMAL	ON or ACTIVE	In operation
INACTIVE	REVERSE	ON or ACTIVE	<u>Not in operation</u>
ACTIVE	REVERSE	OFF or INACTIVE	In operation

Analog Input:

This property, of the type REAL, indicates the current value, in engineering units, of the input being measured. The Present_Value property must be writable when Out_Of_Service is TRUE.

Analog output:

This property, of type REAL, indicates the current value, in engineering units, of the output.

Schedule:

This property indicates the current value of the schedule. This can be any primitive data type. As a result, most analog, binary, and enumerated values may be used. This property must be writable when Out_Of_Service is TRUE (see section "Out_of_Service").

Any change in the value of this property shall be written to all members of the List_Of_Object_Property_References property. An error writing to any member of the list shall not stop the Schedule object from writing to the remaining members.

The normal calculation of the value of the Present_Value property is illustrated as follows (the actual algorithm used is a local matter but must yield the same results as this one):

1. Determine the Exception_Schedule field element using the highest relative priority (as defined in section "Exception_Schedule") that is valid for the current day and whose current value (see method below) is not ZERO. Assign this value to the property Present_Value.
2. If the Present_Value was not assigned in the previous step, then evaluate the current value of the Weekly_Schedule array element for the current day and if that value is not ZERO, assign it to the Present_Value property. If its value is not ZERO, assign the value to the Present_Value property.
3. If the Present_Value was not assigned in the previous steps, then assign the value of the Schedule_Default property to the Present_Value property.

The method for evaluating the current value of a schedule (either Exception or Weekly) is to find the latest element in the list of BACnetTimeValues that occurs on or before the current time. That element's value is used as the current value for the schedule. If no such element is found, then the current value for the schedule is ZERO.

These calculations are such that they can be performed at any time and the correct value of Present_Value property will result. These calculations must be performed at 00:00 each day, whenever the device resets, whenever properties that can affect the results are changed, whenever the time in the device changes by an amount that may have an effect on the calculation result, and at other times, as required, to maintain the correct value of the Present_Value property through the normal passage of time.

Note that the `Present_Value` property is assigned the value of the `Schedule_Default` property at 00:00 of any given day, unless there is an entry for 00:00 in effect for that day. If a scheduled event logically begins on one day and ends on another, an entry at 00:00 is placed in the schedule that is in effect for the second day, and for any subsequent days of the event's duration, to ensure the correct result whenever `Present_Value` is calculated.

16.1.60 `Priority_Array`

Data Type	BACnetPriorityArray
Objects	Binary Output, Analog Output
Description	This property is a read-only array that contains prioritized values/commands that are in effect for this object. See section "Prioritization" for a description of the prioritization mechanism.

16.1.61 `Priority_For_Writing`

Data Type	Unsigned (1...16)
Objects	Schedule
Description	This property defines the priority at which the referenced properties are commanded. It corresponds to the 'Priority' parameter of the WriteProperty service. It is an unsigned integer in the range 1-16, with 1 being considered the highest priority and 16 the lowest.

16.1.62 `Protocol_Object_Types_Supported`

Data Type	BACnetObjectTypesSupported
Objects	Device
Description	This property, of type BACnetObjectTypesSupported, indicates which standardized objects are supported by this device's protocol implementation. The list of properties supported for a particular object may be acquired by use of the ReadPropertyMultiple service with a property reference of ALL.

16.1.63 Protocol_Revision

Data Type	Unsigned
Objects	Device
Description	This property, of the type Unsigned, indicates the minor revision level of the BACnet standard. This value starts at 1 and increases in increments for any substantive change(s) to the BACnet standard that affect device communication or behavior. This value reverts to zero upon each change to the Protocol_Version property. Changes to the values for Protocol_Version and Protocol_Revision are recorded in the History of Revisions at the end of this standard. This property is required for all devices implementing BACnet Protocol_Version 1, Protocol_Revision 1 and above. Absence of this property indicates a device implemented to a version of the standard prior to the definition of the Protocol_Revision property.

16.1.64 Protocol_Services_Supported

Data Type	BACnetServicesSupported
Objects	Device
Description	This property, of the type BACnetServicesSupported, indicates which standardized protocol services are supported by this device's protocol implementation.

16.1.65 Protocol_Version

Data Type	Unsigned
Objects	Device
Description	This property, of type Unsigned, represents the version of the BACnet protocol supported by this BACnet Device. Every major revision of BACnet increases this version number by one. The initial release of BACnet is Version 1.

16.1.66 Read_Only

Data Type	BOOLEAN
Objects	File
Description	This property of the type BOOLEAN indicates whether file data may be changed by a BACnet-AtomicWriteFile service (FALSE) or not (TRUE).

16.1.67 Record_Count

Data Type	Unsigned
Objects	File
Description	This property, of the type Unsigned, indicates the size of the file data in records. The Record_Count property may be present only if File_Access_Type is RECORD_ACCESS. If the number of records can be changed by writing to the file, then this property is writable. Writing to the Record_Count property with a value less than the current size of the file truncates the file at the specified position. Writing a Record_Count of 0 deletes all of the file data but not the File object itself. Writing to the Record_Count property with a value greater than the current size of the file expands the size of the file, but the value of the new octets of the file are determined locally.

16.1.68 Reliability

Data Type	BACnetReliability
Objects	Binary Input, Binary Output, Analog Input, Analog Output, Scheduler
Description	The Reliability property, of the type BACnetReliability, provides an indication of whether the Present_Value or the operation of the physical input/output in question is "reliable" as far as the BACnet Device or operator can determine and, if not, why. The Reliability property for this object may have any of the following values:

Binary Input:

{NO_FAULT_DETECTED, NO_SENSOR, OPEN_LOOP, SHORTED_LOOP, UNRELIABLE_OTHER}

Binary Output:

{NO_FAULT_DETECTED, NO_OUTPUT, OPEN_LOOP, SHORTED_LOOP, UNRELIABLE_OTHER}

Analog Input:

{NO_FAULT_DETECTED, NO_SENSOR, OVER_RANGE, UNDER_RANGE, OPEN_LOOP, SHORTED_LOOP, UNRELIABLE_OTHER}

Analog Output:

{NO_FAULT_DETECTED, OPEN_LOOP, SHORTED_LOOP, NO_OUTPUT, UNRELIABLE_OTHER}

Schedule:

The property Reliability indicates whether the properties of the schedule object are constant. All non-ZERO values used in the Weekly_Schedule, the Exception_Schedule, and the Schedule_Default properties shall be of the same data type, and all members of the List_Of_Object_Property_References shall be writable with that data type. If these conditions are not met, then this property shall have the value CONFIGURATION_ERROR. The Reliability property for this object may have any of the following values:

{NO_FAULT_DETECTED, CONFIGURATION_ERROR, UNRELIABLE_OTHER}

If the List_Of_Object_Property_References contains a member that references a property in a remote device, the detection of a configuration error may be delayed until an attempt is made to write a scheduled value.

16.1.69 Relinquish_Default

Data Type	see Objects
Objects	Binary Output (BACnetBinaryPV), Analog Output (REAL)
Description	This property is the default value to be used for the Present_Value property when all command priority values in the Priority_Array property have a ZERO value.

16.1.70 Resolution

Data Type	REAL
Objects	Analog Input, Analog Output
Description	This property, of the type REAL, indicates the smallest recognizable change in the Present_Value property in engineering units (read-only).

16.1.71 Schedule_Default

Data Type	ANY
Objects	Scheduler
Description	This property contains a default value for the Present_Value property if no other Schedule Value is valid (see section "Present_Value"). This may be a primitive data type. If the Schedule_Default property is written with a value containing a data type not supported by this instance of the Schedule Object (e.g., the List_Of_Object_Property_References property cannot be configured to reference a property of the unsupported data type), the device may return Result(-), specifying an error class of PROPERTY and an error code of DATATYPE_NOT_SUPPORTED.

16.1.72 Segmentation_Supported

Data Type	BACnetSegmentation
Objects	Device
Description	This property, of the type BACnetSegmentation, indicates whether the BACnet Device supports segmentation of messages and, if so, whether it supports segmented transmission, reception, or both: {SEGMENTED_BOTH, SEGMENTED_TRANSMIT, SEGMENTED_RECEIVE, NO_SEGMENTATION}

16.1.73 Status_Flags

Data Type	BACnetStatusFlags
Objects	Binary input, Binary output, Analog input, Analog output, Scheduler
Description	This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a binary or analog input or output or a scheduler object. Three of the flags (for the schedule object two objects) are associated with the values of other properties of this object. A more detailed status could be determined by reading the properties that are linked to these flags. The relationship between individual flags is not defined by the protocol. {IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}

IN_ALARM has the following meaning:

- **Binary Input/Output/Analog Input/Output:**
logical FALSE (0) of the property Event_State has the value NORMAL, otherwise logical TRUE (1).
- **Schedule:**
The value of this flag is logical FALSE (0).

FAULT has the following meaning:

Logical TRUE (1) if the Reliability property is present and does not have a value of NO_FAULT_DETECTED, otherwise logical FALSE (0).

OVERRIDDEN has the following meaning:

- **Binary Input/Analog Input:**
Logical TRUE (1) if the point has been overridden by some mechanism local to the BACnet Device. In this case, "overridden" is taken to mean that the Present_Value and Reliability properties are no longer tracking changes to the physical input. Otherwise, the value is logical FALSE (0).
- **Binary Output/Analog Output:**
Logical TRUE (1) if the point has been overridden by some mechanism local to the BACnet Device. In this case, "overridden" is taken to mean that the properties Present_Value and Reliability no longer represent the physical output. Otherwise, the value is logical FALSE (0).

- **Schedule:**
Logical TRUE (1) if the schedule object has been overridden by some mechanism local to the BACnet Device. In this case, "overridden" is taken to mean that the Present_Value property cannot be changed through BACnet services. Otherwise, the value is logical FALSE (0).

OUT_OF_SERVICE has the following meaning:

Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE (0).

16.1.74 System_Status

Data Type	BACnetDeviceStatus
Objects	Device
Description	This property, of the type BACnetDeviceStatus, reflects the current physical and logical state of the BACnet device. This property can take on the following values: {OPERATIONAL, OPERATIONAL_READ_ONLY, DOWNLOAD_REQUIRED, DOWNLOAD_IN_PROGRESS, NON_OPERATIONAL}

16.1.75 Time_Delay

Data Type	Unsigned
Objects	Binary Input, Binary Output, Analog Input, Analog Output
Description	Binary Input: This property, of the type Unsigned, specifies the minimum period of time in seconds during which the Present_Value must remain equal to the Alarm_Value property before a TO-OFFNORMAL event is generated, or remain not equal to the Alarm_Value property before a TO-NORMAL event is generated. Binary Output: This property, of the type Unsigned, specifies the minimum period of time in seconds during which the Present_Value must be different from the Feedback_Value property before a TO-OFFNORMAL event is generated or must remain equal to the Feedback_Value property before a TO-NORMAL event is generated.

Analog Input, Analog Output:

This property, of the type Unsigned, specifies the minimum period of time in seconds that the Present_Value must remain outside the band defined by the High_Limit and Low_Limit properties before a TO-OFFNORMAL event is generated or within the same band, including the Deadband property, before a TO-NORMAL event is generated.

16.1.76 Time_Of_Active_Time_Reset

Data Type	BACnetDateTime
Objects	Binary Input, Binary Output
Description	This property, of the type BACnetDateTime, represents the date and time at which the Elapsed_Active_Time property was most recently set to a zero value.

16.1.77 Time_Of_State_Count_Reset

Data Type	BACnetDateTime
Objects	Binary Input, Binary Output
Description	This property, of the type BACnetDateTime, represents the date and time at which the Change_Of_State_Count property was most recently set to a zero value.

16.1.78 Units

Data Type	BACnetEngineeringUnits
Objects	Analog Input, Analog Output
Description	This property, of the type BACnetEngineeringUnits, indicates the measurement units of this object.



Information

Additional Information

The list of standardized Engineering Units can be found in the documentation for the BACnet Library on the web site <http://www.wago.com> under Service → Downloads → Building Automation → BACnet Downloads → Software

16.1.79 Update_Interval

Data Type	Unsigned
Objects	Analog Input
Description	This property, of the type Unsigned, indicates the maximum period of time between updates to the Present_Value in hundredths of a second when the input is not overridden and not out-of-service.

16.1.80 UTC_Offset

Data Type	INTEGER
Objects	Device
Description	The UTC_Offset property, of the type INTEGER, indicates the number of minutes (-780 to +780) offset between local standard time and Universal Time Coordinated. The time zones to the west of the zero degree meridian are positive values, and those to the east are negative values. The value of the UTC_Offset property is subtracted from the UTC received in UTCTimeSynchronization service requests to calculate the correct local standard time.

16.1.81 Vendor_Identifier

Data Type	Unsigned16
Objects	Device
Description	This property, of the type Unsigned16, is a unique vendor identification code, assigned by ASHRAE, which is used to distinguish proprietary extensions to the protocol.

16.1.82 Vendor_Name

Data Type	CharacterString
Objects	Device
Description	This property, of the type CharacterString, identifies the manufacturer of the BACnet Device.

16.1.83 Weekly_Schedule

Data Type	BACnetARRAY[7] for BACnetDailySchedule
Objects	Scheduler
Description	This property is a BACnet field with exactly seven elements. Each of the elements 1-7 contains a BACnetDailySchedule. A BACnetDailySchedule consists of a list with BACnetTimeValues, i.e. time/value pairs that indicate the course of planned actions of a weekday, if no Exception_Schedule applies. The field elements 1-7 correspond to the days Monday - Sunday. The Weekly_Schedule is an optional property, but every instance of a Schedule Object must have either a Weekly_Schedule or a non-empty Exception_Schedule. If the Weekly_Schedule property is written with a schedule element containing a data type not supported by this instance of the Schedule Object (e.g., the List_Of_Object_Property_References property cannot be configured to reference a property of the unsupported data type), the device may return Result(-) with error class PROPERTY and an error code of DATATYPE_NOT_SUPPORTED.

16.2 MIB II Groups

16.2.1 System Group

The system group contains general information about the coupler/controller.

Table 274: MIB II – System Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.1.1	sysDescr	R	This entry contains the device identification. The object has a fixed code (e.g., "WAGO 750-841")
1.3.6.1.2.1.1.2	sysObjectID	R	This entry contains the manufacturer's authorization identification.
1.3.6.1.2.1.1.3	sysUpTime	R	This entry contains the time (in hundredths of a second) since the management unit has been last reset.
1.3.6.1.2.1.1.4	sysContakt	R/W	This entry contains the identification and contact information for the system contact person.
1.3.6.1.2.1.1.5	sysName	R/W	This entry contains the administration-assigned device name.
1.3.6.1.2.1.1.6	sysLocation	R/W	This entry contains the node's physical location.
1.3.6.1.2.1.1.7	sysServices	R	This entry designates the quantity of services that this coupler/controller contains.

16.2.2 Interface Group

The interface group contains information and statistics about the device interface.

A device interface describes the Ethernet interface of a coupler/controller and provides status information on the physical Ethernet ports as well as on the internal loopback interface.

Table 275: MIB II – Interface Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.2.1	ifNumber	R	Number of network interfaces in this system
1.3.6.1.2.1.2.2	ifTable	-	List of network interfaces
1.3.6.1.2.1.2.2.1	ifEntry	-	Network interface entry
1.3.6.1.2.1.2.2.1.1	ifIndex	R	This entry contains a unique value for each interface
1.3.6.1.2.1.2.2.1.2	ifDescr	R	This entry contains the name of the manufacturer, the product name, and the version of the hardware interface: e.g., "WAGO Kontakttechnik GmbH 750-841: Rev 1.0"
1.3.6.1.2.1.2.2.1.3	ifType	R	This entry describes the type of interface. ETHERNET CSMA/CD = 6 Software Loopback = 24
1.3.6.1.2.1.2.2.1.4	ifMtu	R	This entry specifies the largest transfer unit; i.e., the maximum telegram length that can be transferred via this interface.
1.3.6.1.2.1.2.2.1.5	ifSpeed	R	This entry indicates the interface speed in bits per second.
1.3.6.1.2.1.2.2.1.6	ifPhysAddress	R	This entry indicates the physical address of the interface. For example, for Ethernet, this entry contains a MAC ID.
1.3.6.1.2.1.2.2.1.7	ifAdmin-Status	R/W	This entry specifies the desired state of the interfaces. Possible values are: up(1): Ready for operation for transmission and reception down(2): Interface is switched off testing(3): Interface is in test mode
1.3.6.1.2.1.2.2.1.8.	ifOperStatus	R	This entry indicates the current operational state of the interface.
1.3.6.1.2.1.2.2.1.9.	ifLastChange	R	This entry indicates the value of the sysUpTime when the state was last changed.
1.3.6.1.2.1.2.2.1.10	ifInOctets	R	This entry gives the total number of bytes received via interface.
1.3.6.1.2.1.2.2.1.11	ifInUcastPkts	R	This entry indicates the number of received unicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.12	ifInNUcastPkts	R	This entry indicates the number of received broad and multicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.13	ifInDiscards	R	This entry indicates the number of packets that were discarded even though no errors had been detected.
1.3.6.1.2.1.2.2.1.14	ifInErrors	R	This entry indicates the number of received packets that contained errors preventing them from being deliverable to a higher layer.

Table 275: MIB II – Interface Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.2.2.1.15	IfInUnknown-Protos	R	This entry indicates the number of received packets sent to an unknown or unsupported port number.
1.3.6.1.2.1.2.2.1.16	ifOutOctets	R	This entry gives the total number of bytes sent via interface.
1.3.6.1.2.1.2.2.1.17	ifOutUcastPkts	R	This entry contains the number of outgoing unicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.18	ifOutNUcastPkts	R	This entry indicates the number of outgoing broad and multicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.19	ifOutDiscards	R	This entry indicates the number of packets that were discarded even though no errors had been detected.
1.3.6.1.2.1.2.2.1.20	ifOutErrors	R	This entry indicates the number of packets that could not be transmitted because of errors.

16.2.3 IP Group

The IP group contains information about IP communication.

Table 276: MIB II – IP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.4.1	ipForwarding	R/W	1: Host is a router; 2: Host is not a router
1.3.6.1.2.1.4.2	ipDefaultTTL	R/W	Default value for the Time-To-Live field of each IP frame.
1.3.6.1.2.1.4.3	ipInReceives	R	Number of received IP frames, including those received in error.
1.3.6.1.2.1.4.4	ipInHdrErrors	R	Number of received IP frames with header errors.
1.3.6.1.2.1.4.5	ipInAddrErrors	R	Number of received IP frames with a misdirected IP address.
1.3.6.1.2.1.4.6	ipForwDatagrams	R	Number of received IP frames passed on (routed)
1.3.6.1.2.1.4.7	ipUnknownProtos	R	Number of received IP frames with an unknown protocol type.
1.3.6.1.2.1.4.8	ipInDiscards	R	Number of received IP frames rejected although no disturbance was present.
1.3.6.1.2.1.4.9	ipInDelivers	R	Number of received IP frames passed on a higher protocol layer.
1.3.6.1.2.1.4.10	ipOutRequests	R	Number of sent IP frames
1.3.6.1.2.1.4.11	ipOutDiscards	R	Number of rejected IP Frames that should have been sent.
1.3.6.1.2.1.4.12	ipOutNoRoutes	R	Number of sent IP frames rejected because of incorrect routing information.
1.3.6.1.2.1.4.13	ipReasmTimeout	R	Minimum time duration until an IP frame is re-assembled.
1.3.6.1.2.1.4.14	ipReasmReqds	R	Minimum number of the IP fragments for building up and passing on.
1.3.6.1.2.1.4.15	ipReasmOKs	R	Number of IP frames re-assembled successfully.
1.3.6.1.2.1.4.16	ipReasmFails	R	Number of IP frames not re-assembled successfully.
1.3.6.1.2.1.4.17	ipFragOKs	R	Number of IP frames fragmented and passed on.
1.3.6.1.2.1.4.18	ipFragFails	R	Number of IP frames that should have been fragmented but could not be, because their don't fragment bit was set in the header.
1.3.6.1.2.1.4.19	ipFragCreates	R	Number of generated IP fragment frames
1.3.6.1.2.1.4.20	ipAddrTable	-	Table of all local IP addresses of the coupler/controller.
1.3.6.1.2.1.4.20.1	ipAddrEntry	-	Address information for an entry
1.3.6.1.2.1.4.20.1.1	ipAdEntAddr	R	The IP address corresponding to the entry's address information
1.3.6.1.2.1.4.20.1.2	ipAdEntIfIndex	R	Index of the interface
1.3.6.1.2.1.4.20.1.3	ipAdEntNetMask	R	The entry's associated subnet mask
1.3.6.1.2.1.4.20.1.4	ipAdEntBcastAddr	R	Value of the last significant bit in the IP broadcast address
1.3.6.1.2.1.4.20.1.5	IpAdEntReasm- MaxSize	R	The size of the longest IP telegram that can be defragmented (reassembled) again.
1.3.6.1.2.1.4.23	ipRoutingDiscards	R	Number of deleted routing entries

16.2.4 IpRoute Table Group

The IP route table contains information about the routing table in the coupler/controller.

Table 277: MIB II – IpRoute Table Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.4.21	ipRouteTable	-	IP routing table
1.3.6.1.2.1.4.21.1	ipRouteEntry	-	A routing entry for a particular destination
1.3.6.1.2.1.4.21.1.1	ipRouteDest	R/W	This entry indicates the destination address of the routing entry
1.3.6.1.2.1.4.21.1.2	ipRouteIfIndex	R/W	This entry indicates the index of the interface, which is the next route destination
1.3.6.1.2.1.4.21.1.3	ipRouteMetric1	R/W	The primary route to the target system
1.3.6.1.2.1.4.21.1.4	ipRouteMetric2	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.5	ipRouteMetric3	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.6	ipRouteMetric4	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.7	ipRouteNextHop	R/W	The IP address of the next route section
1.3.6.1.2.1.4.21.1.8	ipRouteType	R/W	The route type
1.3.6.1.2.1.4.21.1.9	ipRouteProto	R	Routing mechanism via which the route is developed
1.3.6.1.2.1.4.21.1.10	ipRouteAge	R/W	Number of seconds since then the route was last renewed/examined
1.3.6.1.2.1.4.21.1.11	ipRouteMask	R/W	This entry contents the subnet mask for this entry
1.3.6.1.2.1.4.21.1.12	ipRouteMetric5	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.13	ipRouteInfo	R/W	A reference to a special MIB

16.2.5 ICMP Group

Table 278: MIB II – ICMP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.5.1	icmpInMsgs	R	Number of received ICMP messages
1.3.6.1.2.1.5.2	icmpInErrors	R	Number of received ICMP errors containing ICMP-specific errors
1.3.6.1.2.1.5.3	icmpInDestUnreachs	R	Number of received ICMP destination unreachable messages
1.3.6.1.2.1.5.4	icmpInTimeExcds	R	Number of received ICMP time exceeded messages
1.3.6.1.2.1.5.5	icmpInParmProbs	R	Number of received ICMP parameter problem messages
1.3.6.1.2.1.5.6	icmpInSrcQuenchs	R	Number of received ICMP source quench messages
1.3.6.1.2.1.5.7	icmpInRedirects	R	Number of received ICMP redirect messages
1.3.6.1.2.1.5.8	icmpInEchos	R	Number of received ICMP echo request messages (Ping)
1.3.6.1.2.1.5.9	icmpInEchoReps	R	Number of received ICMP echo reply messages (Ping)
1.3.6.1.2.1.5.10	icmpInTimestamps	R	Number of received ICMP timestamp request messages
1.3.6.1.2.1.5.11	icmpInTimestampReps	R	Number of received ICMP timestamp reply messages
1.3.6.1.2.1.5.12	icmpInAddrMasks	R	Number of received ICMP address mask request messages
1.3.6.1.2.1.5.13	icmpInAddrMaskReps	R	Number of received ICMP address mask reply messages
1.3.6.1.2.1.5.14	icmpOutMsgs	R	Number of sent ICMP messages
1.3.6.1.2.1.5.15	icmpOutErrors	R	Number of sent ICMP messages that could not be sent due to errors
1.3.6.1.2.1.5.16	icmpOutDestUnreachs	R	Number of sent ICMP destination unreachable messages
1.3.6.1.2.1.5.17	icmpOutTimeExcds	R	Number of sent ICMP time exceeded messages
1.3.6.1.2.1.5.18	icmpOutParmProbs	R	Number of sent ICMP parameter problem messages
1.3.6.1.2.1.5.19	icmpOutSrcQuenchs	R	Number of sent ICMP source quench messages
1.3.6.1.2.1.5.20	icmpOutRedirects	R	Number of sent ICMP redirection messages
1.3.6.1.2.1.5.21	icmpOutEchos	R	Number of sent ICMP echo request messages
1.3.6.1.2.1.5.22	icmpOutEchoReps	R	Number of sent ICMP echo reply messages
1.3.6.1.2.1.5.23	icmpOutTimestamps	R	Number of sent ICMP timestamp request messages
1.3.6.1.2.1.5.24	icmpOutTimestampReps	R	Number of sent ICMP timestamp reply messages
1.3.6.1.2.1.5.25	icmpOutAddrMasks	R	Number of sent ICMP address mask request messages
1.3.6.1.2.1.5.26	icmpOutAddrMaskReps	R	Number of sent ICMP address mask reply messages

16.2.6 TCP Group

Table 279: MIB II – TCP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.6.1	tcpRtoAlgorithm	R	Retransmission time: 1 = other, 2 = constant, 3 = RSRE, 4 = VANJ
1.3.6.1.2.1.6.2	tcpRtoMin	R	Minimum value for the retransmission timer
1.3.6.1.2.1.6.3	tcpRtoMax	R	Maximum value for the retransmission timer
1.3.6.1.2.1.6.4	tcpMaxConn	R	Number of maximum TCP connections that can exist simultaneously
1.3.6.1.2.1.6.5	tcpActiveOpens	R	Number of existing active TCP connections
1.3.6.1.2.1.6.6	tcpPassiveOpens	R	Number of existing passive TCP connections
1.3.6.1.2.1.6.7	tcpAttemptFails	R	Number of failed connection attempts
1.3.6.1.2.1.6.8	tcpEstabResets	R	Number of connection resets
1.3.6.1.2.1.6.9	tcpCurrEstab	R	The number of TCP connections for which the current state is either Established or Close-Wait
1.3.6.1.2.1.6.10	tcpInSegs	R	Number of received TCP frames including the error frames
1.3.6.1.2.1.6.11	tcpOutSegs	R	Number of correctly sent TCP frames with data
1.3.6.1.2.1.6.12	tcpRetransSegs	R	Number of sent TCP frames retransmitted because of errors
1.3.6.1.2.1.6.13	tcpConnTable	-	For each existing connection, a table entry is created
1.3.6.1.2.1.6.13.1	tcpConnEntry	-	Table entry for connection
1.3.6.1.2.1.6.13.1.1	tcpConnState	R	This entry indicates the status of the TCP connection
1.3.6.1.2.1.6.13.1.2	tcpConnLocalAddress	R	The entry contains the IP address for the connection. For a server, this entry is constant 0.0.0.0
1.3.6.1.2.1.6.13.1.3	tcpConnLocalPort	R	The entry indicates the port number of the TCP connection.
1.3.6.1.2.1.6.13.1.4	tcpConnRemAddress	R	The entry contains the remote IP address of the TCP connection.
1.3.6.1.2.1.6.13.1.5	tcpConnRemPort	R	The entry contains the remote port of the TCP connection.
1.3.6.1.2.1.6.14	tcpInErrs	R	Number of received incorrect TCP frames
1.3.6.1.2.1.6.15	tcpOutRsts	R	Number of sent TCP frames with set RST flag

16.2.7 UDP Group

Table 280: MIB II – UDP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.7.1	udpInDatagrams	R	Number of received UDP frames that could be passed on to the appropriate applications
1.3.6.1.2.1.7.2	udpNoPorts	R	Number of received UDP frames that could not be passed on to the appropriate applications (port unreachable)
1.3.6.1.2.1.7.3	udpInErrors	R	Number of received UDP frames that could not be passed on to the appropriate applications for other reasons.
1.3.6.1.2.1.7.4	udpOutDatagrams	R	Number of sent UDP frames
1.3.6.1.2.1.7.5	udpTable	-	A table entry is created for each application that received UDP frames
1.3.6.1.2.1.7.5.1	udpEntry	-	Table entry for an application that received an UDP frame
1.3.6.1.2.1.7.5.1.1	udpLocalAddress	R	IP address of the local UDP server
1.3.6.1.2.1.7.5.1.2	udpLocalPort	R	Port number of the local UDP server

16.2.8 SNMP Group

Table 281: MIB II – SNMP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.11.1	snmpInPkts	R	Number of received SNMP frames
1.3.6.1.2.1.11.2	snmpOutPkts	R	Number of sent SNMP frames
1.3.6.1.2.1.11.3	snmpInBadVersions	R	Number of received SNMP frames with an invalid version number
1.3.6.1.2.1.11.4	snmpInBadCommunity-Names	R	Number of received SNMP frames with an invalid community
1.3.6.1.2.1.11.5	snmpInBadCommunity Uses	R	Number of received SNMP frames whose community did not have sufficient authorization for the actions that it tried to execute
1.3.6.1.2.1.11.6	snmpInASNParseErrs	R	Number of received SNMP frames with an incorrect structure
1.3.6.1.2.1.11.8	snmpInTooBig	R	Number of received SNMP frames that acknowledged the result too Big
1.3.6.1.2.1.11.9	snmpInNoSuchNames	R	Number of received SNMP frames that acknowledged the result noSuchName
1.3.6.1.2.1.11.10	snmpInBadValues	R	Number of received SNMP frames that acknowledged the result bad value
1.3.6.1.2.1.11.11	snmpInReadOnly	R	Number of received SNMP frames that acknowledged the result readOnly
1.3.6.1.2.1.11.12	snmpInGenErrs	R	Number of received SNMP frames that acknowledged the result genError
1.3.6.1.2.1.11.13	snmpInTotalReqVars	R	Number of received SNMP frames with valid GET or GET-NEXT requests
1.3.6.1.2.1.11.14	snmpInTotalSetVars	R	Number of received SNMP frames with valid SET requests
1.3.6.1.2.1.11.15	snmpInGetRequests	R	Number of GET requests received and processed
1.3.6.1.2.1.11.16	snmpInGetNexts	R	Number of GET-NEXT requests received and processed
1.3.6.1.2.1.11.17	snmpInSetRequests	R	Number of SET requests received and processed
1.3.6.1.2.1.11.18	snmpInGetResponses	R	Number of received GET responses
1.3.6.1.2.1.11.19	snmpInTraps	R	Number of received traps
1.3.6.1.2.1.11.20	snmpOutTooBig	R	Number of sent SNMP frames that contained the result too Big
1.3.6.1.2.1.11.21	snmpOutNoSuchNames	R	Number of sent SNMP frames that contained the result noSuchName
1.3.6.1.2.1.11.22	snmpOutBadValues	R	Number of sent SNMP frames that contained the result bad value
1.3.6.1.2.1.11.24	snmpOutGenErrs	R	Number of sent SNMP frames that contained the result genErrs
1.3.6.1.2.1.11.25	snmpOutGetRequests	R	Number of GET requests sent
1.3.6.1.2.1.11.26	snmpOutGetNexts	R	Number of GET NEXT requests sent
1.3.6.1.2.1.11.27	snmpOutSetRequests	R	Number of SET requests sent
1.3.6.1.2.1.11.28	snmpOutGetResponses	R	Number of GET responses sent
1.3.6.1.2.1.11.29	snmpOutTraps	R	Number of traps sent
1.3.6.1.2.1.11.30	snmpEnableAuthenTraps	R/W	Authentication failure traps(1 = on, 2 = off)

16.3 WAGO MIB Groups

16.3.1 Company Group

The company group contains general information about the company WAGO Kontakttechnik GmbH & Co. KG.

Table 282: WAGO MIB – Company Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.1.1	wagoName	R	Company's registered name Default value: "WAGO Kontakttechnik GmbH & Co. KG"
1.3.6.1.4.1.13576.1.2	wagoDescription	R	Description of company Default value: "WAGO Kontakttechnik GmbH & Co. KG, Hansastr. 27, D-32423 Minden"
1.3.6.1.4.1.13576.1.3	wagoURL	R	URL for company web site Default value: "www.wago.com"

16.3.2 Product Group

The product group contains information about the controller.

Table 283: WAGO MIB – Product Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.1	wioArticleName	R	Name of article Default value: "750-8xx/000-000"
1.3.6.1.4.1.13576.10.1.2	wioArticleDescription	R	Description of article Default value: "WAGO Ethernet (10/100MBit) FBC"
1.3.6.1.4.1.13576.10.1.3	wioSerialNumber	R	Serial number of article Default value: "SNxxxxxxx-Txxxxxx-mac 0030DExxxxxx"
1.3.6.1.4.1.13576.10.1.4	wioMacAddress	R	MAC address of article Default value: "0030DExxxxxx"
1.3.6.1.4.1.13576.10.1.5	wioURLDatasheet	R	URL to datasheet of article Default value: "http://www.wago.com/wagoweb/documentation/navigate/nm0dc__e.htm#ethernet"
1.3.6.1.4.1.13576.10.1.6	wioURLManual	R	URL to manual of article Default value: "http://www.wago.com/wagoweb/documentation/navigate/nm0dc__e.htm#ethernet"

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.7	wioDeviceClass	R	Device class 10 = controller 20 = coupler 30 = switch 40 = display 50 = sensor 60 = actuator
1.3.6.1.4.1.13576.10.1.8	wioDeviceGroup	R	Device group 10 = Serie 750 20 = Serie 758 30 = Serie 767 40 = Serie 762 PERSPECTO

16.3.3 Versions Group

The version group contains information about the hardware/software versions used in the controller.

Table 284: WAGO MIB – Versions Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.10.1	wioFirmwareIndex	R	Index of firmware version
1.3.6.1.4.1.13576.10.1.10.2	wioHardwareIndex	R	Index of hardware version
1.3.6.1.4.1.13576.10.1.10.3	wioFwlIndex	R	Index of software version from firmware loader
1.3.6.1.4.1.13576.10.1.10.4	wioFirmwareVersion	R	Complete firmware string

16.3.4 Real-Time Clock Group

The real-time clock group contains information about the system's real-time clock.

Table 285: WAGO MIB – Real Time Clock Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.11.1	wioRtcDateTime	R/W	Date/time of coupler in UTC as string. For writing date/time use the following string time 11:22:33 date 13-1-2007 Default value: "time xx:xx:xx date xx-xx-xxxx (UTC)"
1.3.6.1.4.1.13576.10.1.11.2	wioRtcTime	R/W	Date/time of coupler in UTC as integer in seconds from 1970-01-01 Default value: "0"
1.3.6.1.4.1.13576.10.1.11.3	wioTimezone	R/W	"Actual time zone of article in hours (-12 - +12) Default value: "0"
1.3.6.1.4.1.13576.10.1.11.4	wioRtcHourMode	R	Hour mode 0 = 12h mode 1 = 24h mode Default value: "0"
1.3.6.1.4.1.13576.10.1.11.5	wioRtcBatteryStatus	R	RTC battery status: 0 = ok 1 = battery empty Default value: "1"
1.3.6.1.4.1.13576.10.1.11.6	wioRtcDayLightSaving	R/W	Time offset of 1 hour: 0 = not offset 1 = offset 1 hour (DayLightSaving) Default value: "0"

16.3.5 Ethernet Group

The Ethernet group contains the settings for the controller on the Ethernet.

Table 286: WAGO MIB – Ethernet Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.12.1	wioEthernetMode	R/W	IP configuration of Ethernet connection: 0 = fix Ip address 1 = dynamic IP address over Bootp 2 = dynamic IP address over DHCP Default value: “1”
1.3.6.1.4.1.13576.10.1.12.2	wioIp	R/W	Actual IP address of coupler
1.3.6.1.4.1.13576.10.1.12.3	wioSubnetMask	R/W	Actual subnet mask of coupler
1.3.6.1.4.1.13576.10.1.12.4	wioGateway	R/W	Actual gateway IP of coupler
1.3.6.1.4.1.13576.10.1.12.5	wioHostname	R/W	Actual host name of coupler
1.3.6.1.4.1.13576.10.1.12.6	wioDomainName	R/W	Actual domain name of coupler
1.3.6.1.4.1.13576.10.1.12.7	wioDnsServer1	R/W	IP address of first DNS server
1.3.6.1.4.1.13576.10.1.12.8	wioDnsServer2	R/W	IP address of second DNS server

16.3.6 Actual Error Group

The actual error group contains information about the last system status/error status.

Table 287: WAGO MIB – Actual Error Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.20.1	wioErrorGroup	R	Error group of last error
1.3.6.1.4.1.13576.10.1.20.2	wioErrorCode	R	Error code of last error
1.3.6.1.4.1.13576.10.1.20.3	wioErrorArgument	R	Error argument of last error
1.3.6.1.4.1.13576.10.1.20.4	wioErrorDescription	R	Error description string

16.3.7 PLC Project Group

The PLC project group contains information about the controller's PLC program.

Table 288: WAGO MIB – PLC Project Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.30.1	wioProjectId	R	ID of CODESYS project
1.3.6.1.4.1.13576.10.1.30.2	wioProjectDate	R	Date of CODESYS project
1.3.6.1.4.1.13576.10.1.30.3	wioProjectName	R	Name of CODESYS project
1.3.6.1.4.1.13576.10.1.30.4	wioProjectTitle	R	Title of CODESYS project
1.3.6.1.4.1.13576.10.1.30.5	wioProjectVersion	R	Version of CODESYS project
1.3.6.1.4.1.13576.10.1.30.6	wioProjectAuthor	R	Author of CODESYS project
1.3.6.1.4.1.13576.10.1.30.7	wioProjectDescription	R	Description of CODESYS project
1.3.6.1.4.1.13576.10.1.30.8	wioNumberOfIecTasks	R	Number of IEC tasks in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9	wioIecTaskTable	-	
1.3.6.1.4.1.13576.10.1.30.9.1	wioIecTaskEntry	-	
1.3.6.1.4.1.13576.10.1.30.9.1.1	wioIecTaskId	R	ID of IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.2	wioIecTaskName	R	Name of IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.3	wioIecTaskStatus	R	Status of IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.4	wioIecTaskMode	R	Mode of IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.5	wioIecTaskPriority	R	Priority of IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.6	wioIecTaskInterval	R	Interval of cyclic IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.7	wioIecTaskEvent	R	Event for IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.8	wioIecTaskCycleCount	R	Count of IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.9	wioIecTaskCycleTime	R	Last cycle time of IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.10	wioIecTaskCycleTime-Min	R	Minimal cycle time of IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.11	wioIecTaskCycleTime-Max	R	Maximal cycle time of IEC task in the CODESYS project
1.3.6.1.4.1.13576.10.1.30.9.1.12	wioIecTaskCycleTime-Avg	R	Average cycle time of IE task in the CODESYS project

16.3.8 Http Group

The Http group contains information and settings for the controller's Web server.

Table 289: WAGO MIB – Http Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.1.1	wioHttpEnable	R/W	Enable/disable the port of the webserver: 0 = port of webserver disable 1 = port of webserver enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.1.2	wioHttpAuthenticationEnable	R/W	Enable/disable the authentication on the websides: 0 = authentication disabled 1 = authentication enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.1.3	wioHttpPort	R/W	Port of the http web server Default value: { 80 }

16.3.9 Ftp Group

The Ftp group contains information and settings for the controller's Ftp server.

Table 290: WAGO MIB – Ftp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.2.1	wioFtpEnable	R/W	Enable/disable the port of the ftp server 0 = port of ftp server disable 1 = port of ftp server enable Default value: { 1 }

16.3.10 Sntp Group

The Sntp group contains information and settings for the controller's Sntp server.

Table 291: WAGO MIB – Sntp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.3.1	wioSntpEnable	R/W	Enable/disable the port of the SNTP server 0 = port of SNTP server disable 1 = port of SNTP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.3.2	wioSntpServer-Address	R/W	IP address of SNTP server Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.3.3	wioSntpClient-Intervall	R/W	Interval to pool SNTP manager Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.3.4	wioSntpClient-Timeout	R/W	Timeout to corrupt SNTP answer Default value: { 2000 }
1.3.6.1.4.1.13576.10.1.40.3.5	wioSntpClient-DayLightSaving	R/W	Time offset of 1 hour: 0 = not offset 1 = offset 1 hour (DayLightSaving) Default value: { 0 }

16.3.11 Snmp Group

The Snmp group contains information and settings for the controller's SNMP agent.

Table 292: WAGO MIB – Snmp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.1	wioSnmpEnable	R/W	Enable/disable the port of the SNMP server 0 = port of SNMP server disable 1 = port of SNMP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.2.1	wioSnmp1-ProtocolEnable	R/W	Enable/disable first SNMPv1/v2c agent Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.2.2	wioSnmp1-ManagerIp	R/W	IP address of first SNMP server Default value: { 'C0A80101'h }
1.3.6.1.4.1.13576.10.1.40.4.2.3	wioSnmp1-Community	R/W	Community identification string for SNMPv1/v2c Default value: { "public" }
1.3.6.1.4.1.13576.10.1.40.4.2.4	wioSnmp1Trap-V1enable	R/W	Enable/disable SNMPv1 traps to first SNMP server Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.2.5	wioSnmp1Trap-V2enable	R/W	Enable/disable SNMPv2c traps to first SNMP server Default value: { 0 }

Table 292: WAGO MIB – Snmp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.2.6	wioSnmp2-ProtocolEnable	R/W	Enable/disable first SNMPv1/v2c agent Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.2.7	wioSnmp2-ManagerIp	R/W	IP address of second SNMP server Default value: { '00000000'h }
1.3.6.1.4.1.13576.10.1.40.4.2.8	wioSnmp2-Community	R/W	Community identification string for SNMPv1/v2c Default value: { "public" }
1.3.6.1.4.1.13576.10.1.40.4.2.9	wioSnmp2Trap-V1enable	R/W	Enable/disable SNMPv1 traps to first SNMP server Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.2.10	wioSnmp2Trap-V2enable	R/W	Enable/disable SNMPv2c traps to first SNMP server Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.3.1	wioSnmp1User-Enable	R/W	Enable/disable first SNMPv3 user Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.2	wioSnmp1-Authentication-Typ	R/W	Authentication typ for first SNMPv3 user: 0 = no Authentication 1 = MD5 Authentication 2 = SHA1 Authentication Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.3	wioSnmp1-Authentication-Name	R/W	Authentication name for first SNMPv3 user Default value: { "SecurityName" }
1.3.6.1.4.1.13576.10.1.40.4.3.4	wioSnmp1-Authentication-Key	R/W	Authentication key for first SNMPv3 user Default value: { "AuthenticationKey" }
1.3.6.1.4.1.13576.10.1.40.4.3.5	wioSnmp1-PrivacyEnable	R/W	Disable/enable data encryption for first SNMPv3 user:: 0 = no Encryption 1 = DES Encryption Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.6	wioSnmp1-PrivacyKey	R/W	Privacy key for SNMPv3 for first SNMPv3 user Default value: { "PrivacyKey" }
1.3.6.1.4.1.13576.10.1.40.4.3.7	wioSnmp1-Notification-Enable	R/W	Enable/disable notification (SNMPv3 traps) with SNMPv3 user Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.8	wioSnmp1-Notification-ReceiverIP	R/W	Receiver IP address for notification (SNMPv3 traps) with SNMPv3 user Default value: { 'C0A80101'h }
1.3.6.1.4.1.13576.10.1.40.4.3.9	wioSnmp2User-Enable	R/W	Enable/disable second SNMPv3 user Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.3.10	wioSnmp2-Authentication-Typ	R/W	Authentication typ for second SNMPv3 user: 0 = no authentication 1 = MD5 authentication 2 = SHA1 authentication Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.11	wioSnmp2-Authentication-Name	R/W	Authentication name for second SNMPv3 user Default value: { "SecurityName" }

Table 292: WAGO MIB – Snmp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.3.12	wioSnmp2-Authentication-Key	R/W	Authentication key for second SNMPv3 user Default value: {"AuthenticationKey"}
1.3.6.1.4.1.13576.10.1.40.4.3.13	wioSnmp2-PrivacyEnable	R/W	Privacy key for SNMPv3 for second SNMPv3 user Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.14	wioSnmp2-PrivacyKey	R/W	Privacy key for SNMPv3 for second SNMPv3 user Default value: { "PrivacyKey" }
1.3.6.1.4.1.13576.10.1.40.4.3.15	wioSnmp2-Notification-Enable	R/W	Enable/disable notification (SNMPv3 traps) with SNMPv3 user Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.3.16	wioSnmp2-Notification-ReceiverIP	R/W	Receiver IP address for notification (SNMPv3 traps) with SNMPv3 user Default value: { '00000000'h }

16.3.12 Snmp Trap String Group

The Snmp trap string group contains strings that are attached to the manufacturer-specific traps.

Table 293: WAGO MIB – Snmp Trap String Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.4.1	wioTrapKbus-Error	R/W	String for 1st SNMP trap Default value: { "Kbus Error" }
1.3.6.1.4.1.13576.10.1.40.4.4.2	wioTrapPlcStart	R/W	String for 2nd SNMP trap Default value: { "Plc Start" }
1.3.6.1.4.1.13576.10.1.40.4.4.3	wioTrapPlcStop	R/W	String for 3rd SNMP trap Default value: { "Plc Stop" }
1.3.6.1.4.1.13576.10.1.40.4.4.4	wioTrapPlc-Reset	R/W	String for 4th SNMP trap Default value: { "Plc Reset" }
1.3.6.1.4.1.13576.10.1.40.4.4.5	wioTrapPlcSoftwareWatchdog	R/W	String for 5th SNMP trap Default value: { "Plc Software Watchdog" }
1.3.6.1.4.1.13576.10.1.40.4.4.6	wioTrapPlc-DivideByZero	R/W	String for 6th SNMP trap Default value: { "Plc Divide By Zero" }
1.3.6.1.4.1.13576.10.1.40.4.4.7	wioTrapPlc-OnlineChange	R/W	String for 7th SNMP trap Default value: { "Plc Online Change" }
1.3.6.1.4.1.13576.10.1.40.4.4.8	wioTrapPlc-Download	R/W	String for 8th SNMP trap Default value: { "Plc Download Programm" }
1.3.6.1.4.1.13576.10.1.40.4.4.9	wioTrapPlc-Login	R/W	String for 9th SNMP trap Default value: { "Plc Login" }
1.3.6.1.4.1.13576.10.1.40.4.4.10	wioTrapPlc-Logout	R/W	String for 10th SNMP trap Default value: { "Plc Logout" }

16.3.13 Snmp User Trap String Group

The Snmp user trap string group contains strings that can be attached to user-specific traps. These strings can be changed via SNMP or Wago_SNMP.lib in CODESYS.

Table 294: WAGO MIB – Snmp User Trap String Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.5.1	wioUserTrapMsg1	R/W	String for 1st SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.2	wioUserTrapMsg2	R/W	String for 2nd SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.3	wioUserTrapMsg3	R/W	String for 3rd SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.4	wioUserTrapMsg4	R/W	String for 4th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.5	wioUserTrapMsg5	R/W	String for 5th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.6	wioUserTrapMsg6	R/W	String for 6th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.7	wioUserTrapMsg7	R/W	String for 7th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.8	wioUserTrapMsg8	R/W	String for 8th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.9	wioUserTrapMsg9	R/W	String for 9th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.10	wioUserTrapMsg10	R/W	String for 10th SNMP trap

16.3.14 Plc Connection Group

Activate or deactivate the connection to CODESYS with the Plc connection group.

Table 295: WAGO MIB – Plc Connection Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.5.1	wioCODESYSEnable	R/W	Enable/disable the port of the CODESYS server 0 = port of CODESYS server disable 1 = port of CODESYS server enable Default value: { 1 }

16.3.15 Modbus Group

The Modbus group contains information and settings about the controller's modbus server.

Table 296: WAGO MIB – Modbus Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.6.1	wioModbusTcp-Enable	R/W	Enable/disable the port of the Modbus TCP server 0 = port of Modbus TCP server disable 1 = port of Modbus TCP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.6.2	wioModbusUdb-Enable	R/W	Enable/disable the port of the Modbus UDP server 0 = port of Modbus UDP server disable 1 = port of Modbus UDP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.6.3	wioMax-Connections	R/W	The maximal count of modbus connections Default value: { 15 }
1.3.6.1.4.1.13576.10.1.40.6.4	wioConnection-Timeout	R/W	Timeout of the modbus connection Default value: { 600 }
1.3.6.1.4.1.13576.10.1.40.6.5	wioModbus-WatchdogMode	R/W	Mode of the modbus watchdog Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.6.6	wioModbus-WatchdogTime	R/W	Timeout of the modbus watchdog Default value: { 100 }
1.3.6.1.4.1.13576.10.1.40.6.7	wioFreeModbus-Sockets	R/W	Unused and free modbus connections Default value: { 15 }
1.3.6.1.4.1.13576.10.1.40.6.8	wioModbus-ConnectionTable	-	
1.3.6.1.4.1.13576.10.1.40.6.8.1	wioModbus-ConnectionEntry	-	
1.3.6.1.4.1.13576.10.1.40.6.8.1.1	wioModbus-ConnectionIndex	R/W	Index of modbus connection
1.3.6.1.4.1.13576.10.1.40.6.8.1.2	wioModbus-ConnectionIp	R/W	IP address of modbus connection
1.3.6.1.4.1.13576.10.1.40.6.8.1.3	wioModbus-ConnectionPort	R/W	Port of modbus connection

16.3.16 Process Image Group

The process image group contains a list of information about the terminals connected to the controller.

Table 297: WAGO MIB – Process Image Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.50.1	wioModulCount	R	Count of modules
1.3.6.1.4.1.13576.10.1.50.2	wioAnalogOutLength	R	Length of analog output process datas
1.3.6.1.4.1.13576.10.1.50.3	wioAnalogInLength	R	Length of analog input process datas
1.3.6.1.4.1.13576.10.1.50.4	wioDigitalOutLength	R	Length of digital output process datas
1.3.6.1.4.1.13576.10.1.50.5	wioDigitalInLength	R	Length of digital input process datas
1.3.6.1.4.1.13576.10.1.50.6	wioDigitalOutOffset	R	Offset of digital output process datas
1.3.6.1.4.1.13576.10.1.50.7	wioDigitalInOffset	R	Offset of digital input process datas
1.3.6.1.4.1.13576.10.1.50.8	wioModuleTable	-	
1.3.6.1.4.1.13576.10.1.50.8.1	wioModuleEntry	-	
1.3.6.1.4.1.13576.10.1.50.8.1.1	wioModuleNumber	R	Number of module slot
1.3.6.1.4.1.13576.10.1.50.8.1.2	wioModuleName	R	Name of module

Table 297: WAGO MIB – Process Image Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.50.8.1.3	wioModuleType	R	Type of module
1.3.6.1.4.1.13576.10.1.50.8.1.4	wioModuleCount	R	Count of module
1.3.6.1.4.1.13576.10.1.50.8.1.5	wioModule- AlternativeFormat	R	Module in alternative format
1.3.6.1.4.1.13576.10.1.50.8.1.6	wioModuleAnalog- OutLength	R	Length of analog output data of module (Bit)
1.3.6.1.4.1.13576.10.1.50.8.1.7	wioModuleAnalog- InLength	R	Length of analog input data of module (Bit)
1.3.6.1.4.1.13576.10.1.50.8.1.8	wioModuleDigital- OutLength	R	Length of digital output data of module (Bit)
1.3.6.1.4.1.13576.10.1.50.8.1.9	wioModuleDigital- InLength	R	Length of digital input data of module (Bit)

16.3.17 Plc Data Group

The Plc data group contains values that can be used for data exchange with CODESYS.

Table 298: WAGO MIB – Plc Data Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.100.1	wioPlcDataTable	-	
1.3.6.1.4.1.13576.10.1.100.1.1	wioPlcDataEntry	-	
1.3.6.1.4.1.13576.10.1.100.1.1.1	wioPlcDataIndex	R/W	Number of plc data DWORD
1.3.6.1.4.1.13576.10.1.100.1.1.2	wioPlcDataReadArea	R/W	Readable plc data (DWORD)
1.3.6.1.4.1.13576.10.1.100.1.1.3	wioPlcDataWriteArea	R	Write-/readable plc data (DWORD)

Glossary

A

ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers)

ASHRAE is an American professional association for HVAC engineering that was founded in 1894.

ASHRAE prepares and publishes manuals, magazines, standards and guidelines for the air-conditioning sector.

The BACnet protocol has been the standard for ASHRAE since 1995 and was accepted by ANSI in 2004 (ANSI/ASHRAE 135-2004).

B

BACnet (Building Automation and Control Networks)

BACnet denotes a fieldbus, a protocol, a standard or a process that can be used to compatibly exchange data between devices of various manufacturers. BACnet was developed in 1987 by ASHRAE for building automation services and has become anchored in different standards since then (ASHRAE/ANSI standard (135-2004), DIN EN ISO standard (16484-5)). BACnet is primarily oriented toward the HVAC sector.

BACnet standard objects and properties, which can be accessed by likewise standardized services, are defined in the standard to achieve the required interoperability.

Manufacturers publish in a document (PICS) those BACnet standard objects, properties and services that they support. This information is provided in blocks broken down into the various functions (BIBBs). Devices are classified in five device profiles to permit easier comparison of the BACnet devices with regard to their various functions and the different possible combinations of objects and services. Each device profile specifies defined minimum requirements for the BACnet functionality that is implemented, facilitating the selection of a suitable device by the customer for specific applications.

BACnet is suitable for the field level as well as the automation level.

BACnet Configurator

The BACnet Configurator is free software that is used for commissioning and configuration of WAGO BACnet/IP controllers. This includes, among other things, logical structuring of the project and network, addressing of the controller, configuration of the client and the service in every WAGO controller and a value browser (for BACnet object properties). The BACnet Configurator is separate, dedicated software and should not be confused with the I/O Configurator for the WAGO-I/O-PRO.

BACnet Data Types

Services utilize standardized BACnet data types and communication units, so-called application layer protocol data units (APDUs) that are defined in an abstract ASN.8824 syntax in accordance with ISO Standard 1 to ensure uniform data transfer.

Elementary data types such as BOOLEAN and INTEGER, and defined BACnet base types are used which consist of SEQUENCES und CHOICES, for example. The latter are made up of elementary or nested, compiled data types.

BACnet/IP

BACnet/IP is a standardized and company-neutral network protocol for building automation services used primarily in the HVAC sector.

This protocol supports building-specific, standard objects and services for integrating devices within networks to ensure interoperability.

This protocol is defined in three (3) layers: BACnet Virtual Link Layer (BVLL) as a backup and bit transfer layer, BACnet Network Layer as the transfer layer and BACnet Application as the application layer. ARCNET, ETHERNET, BACnet/IP, PTP (Point-To-Point) are supported via RS232; MS/TP (Master Slave/Token Passing) via RS485 and LonTalk ANSI/EIA709.1 on the backup and bit transfer layer.

Communication between different devices is executed based on the client/server method, with each device able to act as a client or as a server.

This protocol, which has been an ASHRAE standard since 1995, was accepted as a standard by ANSI (ANSI/ASHRAE 135-2004) and has become anchored in the DIN EN ISO standard 16484-5 (Building Automation Systems, Data Communication Protocols).

BACstac

Function libraries that implement the BACnet protocol and interfaces with well-known high-level languages are known as BACstacs. BACstacs are generally available on the market. BACstacs simplify and accelerate the development of new BACnet devices, as protocol communication at the lowest level is already implemented by BACstac, enabling the developer to build directly upon the application level.

Baseband

Baseband systems are systems that operate without carrier frequencies, i.e. with non-modulated signals. This means they provide exactly one channel that must be logically adapted to the various, specific requirements. Opposite: Broadband

BBMD (BACnet Broadcast Management Device)

BACnet uses broadcast messages ("to everyone") for data communication via networks. Many routers block broadcasts. This is why special routers, BBMDs, are used in networks. A BBMD receives a broadcast that is to be sent to a different network and transmits it directly to the BBMD in the other network. The BBMD receiving the message then transmits the broadcast within its local network. This allows the broadcast to be received by the target device.

B-BC (BACnet-Building-Controller)

The BACnet Building Controller forms one of six different device profiles

described by the BACnet Standard. The B BC is comparable to a control system in building automation services (Direct Digital Control (DDC)).

To meet the requirements of the BACnet Standard for a B-BC, or for a defined device profile, certain BACnet objects, services, etc., must be implemented in the device. These are defined by the function blocks (BIBBs).

The BIBBs that are supported by the device are entered in a document (PICS) that serves as the basis for communication and comparison for manufacturers and customers.

BIBB (BACnet Interoperability Building Block)

A BIBB defines which BACnet features must be implemented in a device for each task. Compared to the BIBBs from different manufacturers, common features represent a basis for interoperability between the devices. BIBBs form the function blocks for the specific interoperability area (IA) and define the functions within this base.

A BIBB is formed from the IA in which it is contained, the Service that can be used and the user (client or server).

Example: DS - RP - A (consisting of IA (DS), Service (RP), User (A))

BIBBs are published in the manufacturers PICS. The Device profile is classified on the basis of the supported BIBBs.

BIG EU (BACnet Interest Group Europe)

The BIG EU is made up of famous members of the building, planning and production industries as well as different training facilities. It publishes reference solutions in the BACnet area to support planners who work in this field. Among other things the NISTIR (National Institute of Standards and Technology Interagency Report) Guideline is also published (<http://www.big-eu.de>).

Bit

Smallest information unit. Its value can either be 1 or 0.

Bit Rate

Number of bits transmitted within a time unit.

BootP

The Bootstrap Protocol sends configuration data to several controllers/computers, etc. (without hard drives). This eliminates the need for manual, individual configuration.

BootP is used at WAGO for assigning IP addresses to couplers/controllers. DHCP reverts back to BootP.

Bridge

A bridge runs on Layer 2 of the ISO/OSI model. Although the bridge corresponds to a Switch, it has only one output, however.

Bridges separate the network into Segments, allowing the number of nodes to be increased. Corrupt data is filtered out. Telegrams are then sent when the target address is located in the linked Segment. Only the frame of the MAC layer is treated. If the destination address is known, the bridge then forwards the data (when the destination address is on a different string than the one where the Frame originated), or destroys it (subscriber already has the frame). If you do not know

the address, the bridge forwards the data in all its known Segments and notes the source address.

A bridge is used to transfer messages independently of the message destination.

Broadband

Transmission technique using a high bandwidth to permit high data transfer rates. This technique allows several devices to transmit simultaneously.

Opposite: Baseband

Broadcast

Broadcast. A message that is sent to all stations connected to the network.

Bus

Bus is a general designation for a line used for bit-parallel or bit-serial data transfer. The bus consists of address, data, control and supply bus. The width of the bus (8, 16, 32, 64-bit) and its frequency are the determining factors for the data transmission rate. The width of the address bus limits network expansion. The fieldbus is a special type of serial bus.

Byte (Binary Yoked Transfer Element)

A data element larger than a bit and smaller than a word. A byte generally contains 8 bits. A byte may contain 9 bits in 36-bit computers.

C

Client

Service-requesting device within the Client Server System. With the aid of the service request, the client can access objects (data) on the Server. The service is provided by the server.

Coaxial Cable

This cable contains one, single conductor and radial shielding for transmitting information.

CSMA/CD (Carrier Sense Multiple Access/Collision Detection)

Random bus access procedure (Carrier Sense Multiple Access with Collision Detection). When a collision is detected, all subscribers back. After waiting (a random delay time), the subscribers attempt to re-transmit the data.

D

Deterministic ETHERNET

Deterministic ETHERNET denotes that the runtimes can be defined and calculated in an ETHERNET network. This is possible by setting up a Switched ETHERNET.

Device profiles (BACnet)

Six (6) device profiles are defined, with each one classifying a minimum number of functions/BIBBs that are supported.

Classification for the devices is specified in Appendix L of the BACnet standards and can be used to determine interoperability in the specific IAs.

Standardized Device Profile:

B-OWS BACnet Operator Workstation

B-BC BACnet Building Controller

B-AAC BACnet Advanced Application Controller

B-SA BACnet Smart Actuator

B-SS BACnet Smart Sensor

B-GW BACnet Gateway

BACnet Gateway Device profiles are published in the manufacturer PICS and make it easier for the customer to compare different BACnet devices with respect to their functions and interoperability.

DHCP (Dynamic Host Configuration Protocol)

This protocol permits automatic configuration of the network for a computer, and also assigns addresses or sets parameters centrally. The DHCP server uses a fixed IP address pool for automatically assigning random, temporary IP addresses to networked computers (Clients) or couplers/controllers, thus saving considerable configuration work in large networks. The client also obtains other information, such as the gateway address (router) and the IP address of the Domain Name System (DNS).

DNS (Domain Name System)

The Domain Name System is a distributed, decentralized database that manages the name sector in the Internet. Unique domain names (such as <http://www.wago.de>) are transformed into IP addresses (such as 123.45.67.123) using a "forward lookup." IP addresses can be converted back to domain names using a "reverse lookup." Using the naming service several IP address can be used for one domain name for distribution of network load. Besides this, domain names are easier to remember than IP addresses. If an IP address changes in the background, this does not affect the domain name. Paul Mockapetris developed the DNS in 1983. Since then it has been expanded to include other standards and has become anchored in the RFC 1034 and RFC.

Driver

Software code, which communicates with a hardware device. This communication is normally performed by internal device registers.

E

EDE (Electronic Data Exchange)

The EDE file serves as a configuration aid for coordinating the functions between devices from different suppliers or commissioning parties. . This file contains the addresses set in the BACnet Configurator and the names of the devices and objects, along with the values for the object properties. Generation of the EDE file for the WAGO BACnet/IP controller can be started directly from the user interface for the BACnet Configurator.

In the "Server" mode the objects for the BACnet/IP controller can be exported into an EDE file. In the "Client" mode, external objects can be imported using the EDE file.

EIB (European Installation Bus)

see KNX

ETHERNET

Specifies a Local Area Network (LAN), which was developed by Xerox, Intel and DEC in the 70's. The bus access process takes place according to the CSMA/CD method.

ETHERNET Standard

ETHERNET was standardized in 1983 with IEEE 802.3 10Base5. ISO accepted this standardization with the ISO Standard 8802/3. ETHERNET can, in the meantime, be used with all common types of cables and with optic fibers. There are, however, some technical and considerable logical differences between the standardized variants and the original "ETHERNET," which is why the term "ETHERNET" is used when the older design is meant and "802.3" is used for standardized systems. The essential differences between the ETHERNET and the IEEE standard are found in the frame architecture and in the handling of pad characters.

F

Fieldbus

The fieldbus is a dedicated bus for the serial transmission of information. Fieldbus systems connect sensors, actuators and controls from the field level to the management level. Numerous different fieldbus systems have been developed for various purposes. For example, the BACnet, LON and KNX fieldbus systems are used primarily in building automation, whereas CANbus and Interbus are applied chiefly in the automotive industry.

Firewall

Collective name for solutions that protect LANs from unauthorized access from the internet. They are also able to control and regulate the traffic from the LAN

into the Internet. The crucial part of firewalls is static Routers, which have an access control list used to decide which data packets can pass from which Host.

Frame

Unit of data transferred at the Data-Link layer. It contains the header and addressing information.

FTP (File Transfer Protocol)

A standard application for TCP/IP, which permits files to be transferred without files being accessed.

Function

Functions are modules that always return the same results (as a function value) when the input values are identical. They have no local variables that store values beyond an invoke.

Function Block

Function blocks are used for IEC 61131 programming and stored in libraries for repeated utilization. A function block is a structured module, which has a name and contains input and output variables, as well as local variables.

G

Gateway

Device for connecting two different networks, performs the translation between differing protocols.

H

Hardware

Electronic, electric and mechanical components of a module.

Header

A portion of the data packet, containing information such as the receiver's address information.

Host

Originally used to describe a central mainframe computer accessed from other systems. The services provided by the subscriber can be called up by means of local and remote request. Today, host can also refer to computers that provide certain services from a central location (such as UNIX hosts on the Internet).

HTML (Hypertext Markup Language)

HTML is the descriptive language for documents on the World Wide Web. It contains language elements for the design of hypertext documents.

HTTP (Hyper Text Transfer Protocol)

Client/Server TCP/IP protocol, which is used on the Internet or Intranets for exchange of HTML documents. It normally uses Port 80.

Hub

A device, which allows communication between several network users via twisted pair cable. Its topology is star-shaped.

HVAC (Heating, Ventilation and Air Conditioning)

HVAC is a special sector of building automation services.

Hypertext

Document format used by HTTP. Hypertext documents are text files that provide links to other text documents via particular highlighted keywords.

I

ICMP (Internet Control Message Protocol)

ICMP is a protocol for transmission of status information and error messages of the IP, TCP and UDP protocols between IP network nodes. ICMP offers, among other things, the possibility of an echo (ping) request to determine whether a destination is available and is responding.

IEC 61131-3

International standard for modern systems with PLC functionality created in 1993. Based on a structured software model, it defines a series of powerful programming languages to be utilized for different automation tasks.

AWL (statement list - STL), ST (structured text), AS (process structure), FUP (function plan), KOP (contact plan).

IEEE

Institute of Electrical and Electronic Engineers.

IEEE 802.3

IEEE 802.3 is an IEEE standard. ETHERNET only supports the yellow cable as a medium (Thicknet ETHERNET coaxial cable). IEEE 802.3 also supports S UTP and Broadband coaxial cable. The segment lengths range from 500 m for yellow cable, 100 m for TP and 1800 m for Broadband coaxial cable. A star or bus topology is possible. ETHERNET (IEEE 802.3) uses CSMA/CD as a channel access method.

Intel Format

Set configuration for the coupler/controller for setting up the process image. In the coupler/controller memory, the module data is aligned in different ways, depending on the set configuration (Intel/Motorola-Format, word alignment, etc.). The format determines whether or not high and low bytes are changed over – they are not changed with the Intel format.

Internet

A collection of networks interconnected to each other throughout the world. It is most commonly referred to as the World Wide Web, or simply as the Web.

Interoperability

Interoperability is the capability of different devices, systems, methods or even organizations to find a common language for mutually achieving a set goal. Systems and software operate with interoperability, for example, when they are linked via interfaces, when they use common network protocols and data formats, or when they contain the same standards. These conditions enable information and data to be exchanged and provided in an efficient manner.

Interoperability Area (IA)

To evaluate the overall interoperability of a system the requirements made on the individual devices are broken down into seven areas (Interoperability Areas) according to their functions. These then serve as the basis for evaluating the interoperability.

The following IAs are defined: Data Sharing (DS), Alarm and Event Notification (AE), Scheduling, (SCHED), Trending (T), Device Management (DM), Network Management, Network Management (NM) und Virtual Terminal Management (VT).

Each of these IAs possesses a collection of interoperability or function modules, the BIBBS. These modules define all services that a client or server can perform with the IA.

The abbreviation for the IA (e.g. DS) is placed in front of the BIBB name.

Intranet

A private network within an organization that allows users to exchange data within that particular organization.

I/O Configurator

The I/O Configurator is a plug-in incorporated into WAGO-I/O-PRO for easier assignment of addresses and protocols for modules at a coupler/controller.

IP (Internet Protocol)

Internet protocol is a network protocol that performs packet-oriented, connectionless and non-acknowledged transfer of data within a network. This protocol builds upon the transfer layer for the ISO/OSI model. Stations identify themselves using IP addresses.

IP Message Tunneling

In addition to BACnet/IP, IP message tunneling is a method for transferring BACnet messages to a network.

Communication via IP message tunneling takes place via the BACnet tunneling router (BTR), which is also designated as "Annex H Router" on account of its description in the annex of the BACnet Standard.

Implementation between the different communication technologies BACnet and IP is performed using routing tables, with a combination of BACnet network numbers and IP addresses. The BACnet protocol requires an Annex H router in both local area networks in order to send a message from one device to another one on a different network. The Annex H router for the first network transfers the

BACnet message to a UDP (User Datagram Protocol) frame and transmits the message over standard IP links, or over the Internet to the Annex H router in network 2.

This router then "unpacks" the incoming data packet and sends the message to the destination device via the BACnet protocol.

BACnet devices do not require IP compatibility for IP message tunneling via BTRs. BTRs are frequently used in existing BACnet networks that have a link to IP networks, to an intranet or to the Internet.

ISO/OSI (Open Systems Interconnection) Model

The ISO/OSI model is a reference model for networks, with the goal of creating open communication. It defines the interface standards of the respective software and hardware requirements between computer manufacturers. The model treats communication removed from specific implementations, using seven layers. The model treats communication removed from specific implementations, using seven layers: 1 -Bit transfer layer, 2 -Backup layer, 3 -Transfer layer, 4 -Transport layer, 5 -Session layer, 6 -Presentation layer and 7 -Application layer.

K

KNX

KNX has been established as a flexible bus system for building automation and has been standardized by the KNX Association in ISO/IEC 14543. KNX was developed by the European installation bus (EIB), BatiBUS and European Home Systems (EHS). In addition to twisted pair, other transmission media, such as powerline, radio and links to ETHERNET ("KNXnet/IP") are also supported.

L

LAN (Local Area Network)

A LAN is a spatially limited, local network for permanently linking computers over shorter distances. Data transfer can take place via ETHERNET, Token Ring and FDDI, as well as wireless (WLAN).

Library

Collection of Modules available to the programmer in the WAGO-I/O-PRO programming tool for creating control programs in accordance with IEC61131-3.

LON (Local Operating Network)

LON is used as a fieldbus for building automation. It was developed 1990 by Echelon and enables, as does KNX, the communication between different devices, independent of the manufacturer and active application.

M

MIB (Management Information Base)

MIB is a collection of information about all parameters, which can be handed over to the management software with a request via SNMP. This enables remote maintenance, monitoring and control of networks to be performed via SNMP protocol.

MODBUS

MODBUS is an open protocol based on the Master/Slave principle. The MODBUS links the master with several clients, via either serial interface or ETHERNET.

Three data transmission types are available: MODBUS/RTU (binary data transmission), MODBUS/TCP (data transmission using TCP/IP packets) and MODBUS/ASCII (ASCII code transmission).

Module

Modules consist of functions, function blocks and programs.

Every module is made up of a declaration part and a body. The body is written in one of the IEC programming languages AWL (statement list - STL), ST (structured text), AS (process structure), FUP (function plan) or KOP (contact plan).

N

Natives BACnet

"Native" BACnet objects are those objects that are recognized by BACnet/IP controllers during run-up, without a configuration being loaded. These objects are created automatically for the plugged binary and analog input and output modules. The configuration for the native objects is loaded from the firmware and an internal database.

Besides the BACnet/IP controller, no additional hardware is necessary for integrating BACnet into a network. All requisite, native BACnet objects, properties and services are directly available. This standard behavior for a BACnet/IP controller can be deactivated in the BACnet Configurator, or in the I/O Configuration using the option "Disable native BACnet".

A gateway is required for communication with non-native BACnet devices/networks.

Complex modules do not have any direct equivalent as BACnet objects. Objects for these modules must be created and configured using the WAGO-I/O-PRO I/O Configurator.

O

Object

BACnet communication takes place using standardized objects (DIN EN 16484-

5). These objects are tailored specifically to the HVAC sector for building automation services. Both simple field devices as well as complex automation control systems can be modeled in the object presentation. Objects may represent both physical inputs and outputs and virtual objects, such as counting values. Objects possess certain properties, whose values can be read and/or written using services.

All of the objects supported by a device are also described in the PICS data from the specific manufacturer.

Open MODBUS/TCP Specification

Specification, which establishes the specific structure of a MODBUS/TCP data packet. This is dependent upon the selected function code or upon the selected function (import or export bit or register).

Operating System

An operating system is software used for managing equipment such as memory and connected devices, and for executing programs.

P

Parameter Setting

Parameterization is defined as the assignment and storage of set-up and configuration data as they are required for the execution of predefined functions.

Ping Command

When a ping command (ping < IP address) is entered, the ping program ICMP generates echo request packets. It is used to test for node availability.

Port Number

The port number, in conjunction with the IP address, is the unique connection point between two processes (applications).

Predictable ETHERNET

The predictable message delay time on an ETHERNET network. The measures that are taken here enable nearly real-time requests to be realized.

Prioritization

BACnet applications can access Objects through Services and change their properties. Access prioritization is required to regulate and organize access of various applications to the properties. The BACnet standard distinguishes between 16 different priority levels. A priority can only be set for the current values (Present_Value) of output objects. For this reason they are also called "command properties". The processing sequences for individual applications can be changed using prioritization. Simultaneous access of several applications to the same Objects is regulated in this manner. The application with the highest priority level (lowest number) is given priority.

Property

Objects are described by specific properties and values. In this manner, object information, such as name, status and behavior of an object can be read.

Properties may be editable and readable (R), readable and writable (W) and optionally readable and/or writable (O). Access to object properties is gained using Services.

The properties Object_Identifier, Object_Name, Object_Type and Present_Value are common to all Objects. Other properties are object-specific, depending on the function.

Protocol Implementation Conformance Statement (PICS)

The Protocol Implementation Conformance Statement (PICS) described objects and functions supported by BACnet devices. This is a standard document that must be filled in by the manufacturer. When linking different systems, the PICS and the BIBBs for the devices contained therein are compared with one another.

Proxy Server

Proxy means agent or representative. A proxy server (or proxy gateway) allows indirect access to the network by systems, which do not have direct access to the Internet. This may be systems that are restricted from direct access by a firewall for security reasons. A proxy can filter out individual data packets between the Internet and a local network LAN to enhance security. Proxies are also used to limit access to particular servers.

In addition, proxy servers can also have a cache function. In this case they check whether the respective URL address is already available locally and return it immediately, if necessary. This saves time and costs associated with multiple accesses. If the URL is not in the cache, the proxy forwards the request as normal. The user should not notice the proxy server apart from the single configuration in the web browser. Most web browsers can be configured so that they use different or no proxy gateways per access method (FTP, HTTP).

R

Repeater

Repeaters operate (like hubs, but with only one, instead of several outputs) on Layer 1 of the ISO/OSI model.

Repeaters are physical amplifiers without their own processing function. They refresh data without detecting damaged data and forward all signals. Repeaters are used for implementing greater transmission distances, or when the maximum number of nodes of (normally) 64 devices for each twisted-pair segment is exceeded. The repeater is then always counted as a node in a segment when the maximum number of nodes is reached.

The media can also be changed when routers are used that are configured as repeaters.

Request

A service request from a client, which requests the provision of a service from a server.

Response

Response from a server in reply to a request from a client.

RFC Specifications

Specifications, suggestions, ideas and guidelines regarding the Internet are published in the form of RFCs (Request For Comments).

RJ-45 Connector

The RJ-45 plug is also referred to as a Western connector. This connector creates the connection of two network controllers via a twisted-pair cable.

Router

Routers are used to connect neighboring subnets, with the router operating with addresses and protocols of the third ISO/OSI layer. As this layer is hardware independent, the routers allow transition to another transmission medium. To transmit a message the router evaluates the logical address (source and destination address) and finds the best path if there are several possibilities. Routers can be operated in the Repeater or Bridge modes.

Routing

Method of selecting the best path for sending data to a distant network.

RS-232

The RS-232 (official designation ANSI/EIA/TIA-232-F-1997) is a serial interface for point-to-point connections. "RS" stands for "Radio Sector", but is frequently translated as "Recommended Standard".

Data is transmitted via this interface bit-serial over a data line and received on a different data line. As only one data line is used at a time, the data is transmitted time-delayed consecutively and asynchronously. The time intervals between transmissions of the data may be randomly long.

As the RS-232 interface is a so-called powered interface, data is transferred electrically encoded.

The cable connections used for this interface are 9-pin Sub-D connectors and 9-pin jacks. A distinction is made here between data lines (RxD, TxD and GND) and control lines (DCD, DTR, DSR, RTS, CTS and RI).

S

SCADA (Supervisory Control and Data Acquisition)

SCADA software is a program for the control and visualization of processes (supervisory control and data acquisition).

Segment

Typically, a network is divided up into different physical network segments by way of routers or repeaters.

Server

Service-supplying device within a Client Server System. The service is requested by the Client.

Service

A service is an operation (Read, Write) oriented toward an object. BACnet specified standardized services that are transmitted via a special BACnet network layer. Communication is based on the client/server method: The client makes requests to the server. The server processes the requests from the client and then transmits a response. A distinction is made between 38 services that are broken down into 5 categories: Alarm and events, object access, file access, remote device access, and virtual terminal services

Service port

The service port is located next to the mode switch, behind the cover flap on the controller. This port acts as the configuration and programming interface and is used for communication with *WAGO-I/O-CHECK*, *WAGO-I/O-PRO* and for downloading firmware. A special programming cable (750-920) is necessary.

SMTP (Simple Mail Transfer Protocol)

Standard protocol, with which E-mails are sent via Internet.

SNMP (Simple Network Management Protocol)

SNMP is used for remote maintenance of servers. This allows routers, for example, to be configured directly from the network provider's office, without having to physically visit the customer.

SNTP (Simple Network Time Protocol)

This connectionless network protocol performs time synchronization in networks with a time server via Internet. SNTP is a simplified version of the NTP protocol. On account of this simplification (also with regard to the software), SNTP operates somewhat less exact than NTP. SNTP is defined in RFC 4330.

Socket

A software interface implemented with BSD-UNIX for inter-process communication. Sockets are also possible in the network via TCP/IP. Per Windows 3.11, they are also available in Microsoft operating systems.

STP (Shielded Twisted Pair)

An STP cable is a symmetrical cable with shielded cores twisted in pairs. The classic STP cable is a multi-core cable, whose stranded conductors are isolated. The conductors of the STP cable are individually protected; it has no total shielding.

S-STP (Screened/Shielded Twisted Pair)

In addition to STP cables, S-STP cables are provided with total shielding consisting of foil or network shielding in addition to the single shielding for the individual conductors.

Structured Cabling

With structured cabling, maximum permissible cable lengths are defined (EIA/TIA 568, IS 11801) for site, building and floor cabling, with recommendations for topologies also indicated.

Subnet

A portion of a network that shares the same network address as the other portions. These subnets are distinguished through the subnet mask.

Subnet Mask

Subnet masks can be used to manipulate the address ranges in the IP address area in reference to the number of subnets and hosts. A standard subnet mask, for example, is 255.255.255.0.

S-UTP (Screened Unshielded Twisted-Pair)

Screened twisted pair cable, which only has one external shield. However, the twisted pair cables are not shielded from each other.

Switch

Switches are comparable to bridges, but with several outputs. Each output uses the full ETHERNET bandwidth. Each output uses the full ETHERNET bandwidth. Switches learn which nodes are connected and filter the information transmitted over the network accordingly. Switches learn which nodes are connected and filter the information transmitted over the network accordingly.

Switched ETHERNET

ETHERNET network set up using switches. There are a multitude of applications for switching technologies. ETHERNET switching is becoming increasingly popular in local networks as it allows deterministic ETHERNET.

T

TCP (Transport Control Protocol)

TCP is a connection-oriented network protocol for the transport layer (Layer 4) of the ISO/OSI model provided with relatively secure transmission mechanisms.

TCP/IP Protocol Stack

The TCP/IP protocol stack denotes network protocols that enable communication between different networks and topologies.

Telnet

The Telnet protocol fulfils the function of a virtual terminal. It allows remote access from the user's computer to other computer systems on the network.

Traps

Traps are unsolicited messages, which are sent by an "agent" to a management system, as soon as something unexpected and interesting for the management system happens. Traps are comparable to interrupts from hardware. A well-known example of a trap message is the "Blue screen" with Win95/98.

Twisted Pair

Twisted pair cables (abbreviated to TP).

U

UDP (Users Datagram Protocol)

The user datagram protocol is a communication protocol between two computers and an alternative to TCP (Transmission Control Protocol). As with TCP, UDP communicates via Internet Protocol, although it is somewhat less reliable due to its uncontrolled communication method.

URL (Uniform Resource Locator)

Address form for Internet files which are mostly applied within the World Wide Web (WWW). The URL format makes a unique designation of all documents on the internet possible. It describes the address of a document or objects that can be read by a web browser. URL includes the transmission type (HTTP, FTP), the computer that contains the information and the path on the computer. A URL has the following format:

Document type://Computer name/List of contents/File name.

UTP (Unshielded Twisted Pair)

The UTP cable is a symmetrical, non-protected cable with twisted colored wires in pairs. This cable type, either of a two-pair or four-pair design, is the cable type most used for floor wiring and terminal wiring.

V

Vendor ID (BACnet)

The BACnet standard defines numerical identifiers for suppliers. These IDs can be obtained free of charge from ASHRAE and are a prerequisite for development, marketing and operation of BACnet components in a standard BACnet network. The vendor ID for WAGO Kontakttechnik GmbH & Co. KG is "222". A list of all assigned vendor IDs is available at the following Internet page:
<http://www.bacnet.org/VendorID/index.html>

W

WAGO-I/O-PRO (CODESYS Automation Alliance)

Uniform programming environment, programming tool by WAGO Kontakttechnik GmbH & Co. KG for the generation of a control program as per IEC61131-3 for all programmable fieldbus controllers (PFC). The software enables a program to be created, tested, debugged and started up. The predecessor to the WAGO-I/O-PRO software is the WAGO-I/O-PRO 32, Versions 2.1 and 2.2. The new WAGO-I/O-PRO consists of the basic tool "CODESYS 2.3 CAA" and the target files with the WAGO-specific Drivers.

Web Browser

A Web browser is a program used for reading Hypertext. The browser allows the various documents to be viewed in Hypertext and navigation between documents.

Word Alignment

Set configuration for the fieldbus coupler/controller for setting up the process image. Word-alignment is used to establish the process image word-by-word (2 bytes).

WWW (World Wide Web)

The World Wide Web (network) is a Hypertext system that can be called up via Internet. It is based on the HTTP network protocol, the descriptive language HTML and URLs for unique page (site) addressing.

Literature List



Information

Switching Technology in the Local Network

Mathias Hein

Thomson Executive Press

ISBN 9781850321668

TCP/IP 2: Running a Successful Network

(Data Communications and Networks Series)

Kevin Washburn, Jim Evans

Addison-Wesley Publishing Company, 1996

ISBN 9780201877113

Local Area Networks - An Introduction to the Technology

John E. McNamara

Digital Press, 1995

ISBN 9780135330500

Information on the Internet:

Ready in one day for TCP/IP sockets**Last updated July 2012**

Wiesemann & Theis GmbH

<http://www.wut.de>

BIG EU (BACnet Interest Group Europe)**Last updated July 2012**

The BIG-EU publishes reference solutions in the BACnet area to support planners who work in this field. Among other things the NISTIR (National Institute of Standards and Technology Interagency Report) Guideline is also published.

<http://www.big-eu.de/eng>

ASHRAE (American Society of Heating, Refrigeration and Air-Conditioning Engineers)**Last updated July 2012**

Professional association for all employed in the heating, cooling, ventilation and air-conditioning fields in the USA

<http://www.ashrae.org>

ASHRAE SSPC 135 Committee**Last updated July 2012**

The "Standing Standard Project Committee 135" steers the ongoing development of the protocol

<http://www.bacnet.org>

List of Figures

Figure 1: Fieldbus Node (Example)	22
Figure 2: Marking Area for Serial Numbers	23
Figure 3: Fuse Protection of the Electronic Circuit Power Supply	26
Figure 4: Isolation for Fieldbus Couplers/Controllers (Example).....	27
Figure 5: System Supply via Fieldbus Coupler/Controller (left) and via Internal System Supply Module (right).....	28
Figure 6: System Voltage for Standard Couplers/Controllers and Extended ECO Couplers	29
Figure 7: Field Supply for Standard Couplers/Controllers and Extended ECO Couplers	32
Figure 8: Supply Module with Fuse Carrier (Example 750-610)	34
Figure 9: Removing the Fuse Carrier	35
Figure 10: Opening the Fuse Carrier	35
Figure 11: Changing the Fuse	35
Figure 12: Fuse Modules for Automotive Fuses, Series 282	36
Figure 13: Fuse Modules for Automotive Fuses, Series 2006	36
Figure 14: Fuse Modules with Pivotal Fuse Carrier, Series 281	36
Figure 15: Fuse Modules with Pivotal Fuse Carrier, Series 2002	36
Figure 16: Power Supply Concept.....	37
Figure 17: Supply Example for Standard Couplers/Controllers	38
Figure 18: Carrier Rail Contact (Example)	42
Figure 19: Cable Shield at Ground Potential.....	43
Figure 20: Examples of the WAGO Shield Connecting System.....	44
Figure 21: Application of the WAGO Shield Connecting System	44
Figure 22: View BACnet/IP Controller.....	49
Figure 23: Device Supply	51
Figure 24: RJ-45 Connector	52
Figure 25: Display Elements	53
Figure 26: Service Interface (Closed and Opened Flap)	55
Figure 27: Mode Selector Switch (Closed and Open Damper of the Service Port)	56
Figure 28: Address Selection Switch (for example setting "0")	58
Figure 29: Opening the Memory Card Slot, Inserting an SD/SDHC Memory Card	59
Figure 30: Spacing.....	90
Figure 31: Release Tab of Extended ECO Fieldbus Coupler (Example).....	93
Figure 32: Insert I/O Module (Example).....	94
Figure 33: Snap the I/O Module into Place (Example)	94
Figure 34: Removing the I/O Module (Example)	95
Figure 35: Data Contacts	96
Figure 36: Example for the Arrangement of Power Contacts	97
Figure 37: Connecting a Conductor to a CAGE CLAMP®	98
Figure 38: Run-up of the Controller.....	100
Figure 39: Example of Process Image for Input Data	103
Figure 40: Example of Process Image for Output Data	104
Figure 41: Memory Areas and Data Exchange	107
Figure 42: Example Declaration of Remanent Flags by "VAR RETAIN"	109
Figure 43: Data Exchange Between MODBUS Master and I/O Modules.....	115

Figure 44: Data Exchange Between PLC Function (CPU) of the PFC and the I/O Modules.....	116
Figure 45: Example of Addressing for a Fieldbus Node.....	119
Figure 46: Restore Function.....	122
Figure 47: WBM Page “Backup & Restore”.....	124
Figure 48: WBM Page “Backup & Restore”.....	125
Figure 49: WBM Page “Backup & Restore”.....	126
Figure 50: WBM Page “BACnet”.....	128
Figure 51: Setting the Storage Location in the Web-based Management System.....	132
Figure 52: Logging into the PLC Browser.....	133
Figure 53: Example of Directory Structure on the SD Card.....	136
Figure 54: Address Selection Switch, for Example the Value Setting “50” ($2^1 + 2^4 + 2^5$).....	142
Figure 55: WBM Page “Port”.....	151
Figure 56: Example for the Function test of a Fieldbus Node.....	154
Figure 57: Example of Time Synchronization in ETHERNET Settings.....	159
Figure 58: Example of WBM Clock Configuration.....	160
Figure 59: Logging in for Programming Access.....	162
Figure 60: Logging Into the PLC Browser.....	163
Figure 61: Dialog Window for Target System Settings.....	164
Figure 62: Write Access Over Module Parameters.....	166
Figure 63: EA-config.xml.....	168
Figure 64: Logging in for Programming Access.....	171
Figure 65: Updating the BACnet Image.....	171
Figure 66: Watchdog Runtime is Less Than the Task Runtime.....	179
Figure 67: Watchdog Runtime Is Greater Than the Task Call Interval.....	179
Figure 68: Enabling/Disabling System Events.....	182
Figure 69: Dialog Window “Communication Parameters”.....	187
Figure 70: Logging in for Programming Access.....	188
Figure 71: WBM Page "Information".....	193
Figure 72: WBM Page “Ethernet”.....	195
Figure 73: WBM Page “TCP/IP”.....	198
Figure 74: WBM Page “Port”.....	200
Figure 75: WBM page “SNMP”.....	203
Figure 76: WBM page “SNMP V3”.....	205
Figure 77: WBM Page “Watchdog”.....	207
Figure 78: WBM Page “Clock”.....	210
Figure 79: WBM Page "Security".....	213
Figure 80: WBM Page “Modbus”.....	215
Figure 81: WBM Page “BACnet”.....	217
Figure 82: WBM Page "PLC Info".....	219
Figure 83: WBM Page “PLC Settings”.....	220
Figure 84: WBM Page “Features”.....	225
Figure 85: WBM Page “I/O config” (View 1).....	227
Figure 86: WBM Page “I/O config” (View With Process Values).....	228
Figure 87: WBM Page “Disk Info”.....	230
Figure 88: WBM Page “SD Card”.....	231
Figure 89: WBM Page “Backup & Restore”.....	233
Figure 90: WBM Page “WebVisu”.....	237

Figure 91: Display Elements	238
Figure 92: Node Status – I/O LED Signaling.....	241
Figure 93: Error Message Coding	241
Figure 94: Function Block for Determining Loss of Fieldbus, Independently of Protocol	252
Figure 95: Examples of a Prioritization	278
Figure 96: Example BACnet Interoperability Building Block.....	282
Figure 97: Connection between BACnet Components	286
Figure 98: Use of BACnet in Building Automation.....	287
Figure 99: BACnet Layers in the ISO/OSI Model	288
Figure 100: Communication over an Annex-H router	290
Figure 101: Communication via BACnet/IP	291
Figure 102: Sending a Broadcast Report.....	291
Figure 103: Registration of a Foreign Device	292
Figure 104: Use of the MODBUS Functions	296
Figure 105: Example SCADA Software with MODBUS Driver.....	368
Figure 106: Side Marking Example for Approved I/O Modules According to ATEX and IECEX	370
Figure 107: Text Detail – Marking Example for Approved I/O Modules According to ATEX and IECEX.	370
Figure 108: Side Marking Example for Approved Ex i I/O Modules According to ATEX and IECEX.	372
Figure 109: Text Detail – Marking Example for Approved Ex i I/O Modules According to ATEX and IECEX.	372
Figure 110: Side Marking Example for I/O Modules According to NEC 500 ...	375
Figure 111: Text Detail – Marking Example for Approved I/O Modules According to NEC 500.....	375

List of Tables

Table 1: Variants	13
Table 2: Number Notation.....	16
Table 3: Font Conventions	16
Table 4: Legend for Figure “System Supply via Fieldbus Coupler/Controller (left) and via Internal System Supply Module (right)”	28
Table 5: Alignment.....	29
Table 6: Legend for Figure “Field Supply for Standard Couplers/Controllers and Extended ECO Couplers”	32
Table 7: Power Supply Modules	34
Table 8: Filter Modules for 24 V Supply	37
Table 9: Legend for Figure “Supply Example for Fieldbus Coupler/Controller”. ..	39
Table 10: WAGO Ground Wire Terminals	41
Table 11: Legend for Figure “View BACnet/IP Controller 750-831”	50
Table 12: RJ-45 Connector and RJ-45 Connector Configuration	52
Table 13: Display Elements Fieldbus Status.....	53
Table 14: Display Elements Node Status	53
Table 15: Display Elements Supply Voltage	53
Table 16: Legend for Figure “Service Interface (Closed and Opened Flap)”	55
Table 17: Legend for Figure „Mode Selector Switch“.....	56
Table 18: Mode Selector Switch Positions, Static Positions on PowerOn/Reset..	57
Table 19: Mode Selector Switch Positions, Dynamic Positions During Ongoing Operation.....	57
Table 20: Meaning of DIP Switch Positions	58
Table 21: Technical Data – Device Data.....	61
Table 22: Technical Data – System Data	61
Table 23: Technical Data – Supply	62
Table 24: Technical Data – Fieldbus MODBUS/TCP	62
Table 25: Technical Data – Accessories	63
Table 26: Technical Data – Field Wiring.....	63
Table 27: Technical Data – Power Jumper Contacts	63
Table 28: Technical Data – Data Contacts.....	63
Table 29: Technical Data – Climatic Environmental Conditions	63
Table 30: Technical Data – Mechanical Strength	64
Table 31: BACnet Approvals	65
Table 32: BIBBs of the B-BC	68
Table 33: Properties of the Analog Input Object	79
Table 34: Properties of the Analog Output Object.....	80
Table 35: Properties of the Binary Input Object	81
Table 36: Properties of the Binary Output Object.....	82
Table 37: Properties of the Calendar Object	83
Table 38: Properties of the Device Object	84
Table 39: Properties of the File Object	85
Table 40: Properties of the Schedule Object	86
Table 41: WAGO DIN Rails	90
Table 42: Data Width for I/O Modules	111
Table 43: Example of Addressing	111
Table 44: IEC-61131-3 Address Areas	113
Table 45: Absolute Addressing	113

Table 46: Addressing Example	114
Table 47: Allocation of Digital Inputs and Outputs to Process Data Words in Accordance with the	115
Table 48: Possible Errors on System Start	123
Table 49: Possible Errors During the Backup Function.....	125
Table 50: Possible Errors During the Restore Function.....	127
Table 51: Possible Errors when Inserting the Memory Card During Ongoing Operation.....	130
Table 52: Possible Errors when Removing the Memory Card During Ongoing Operation.....	131
Table 53: Possible Errors During SPS Access to the File System of the Memory Card.....	134
Table 54: Possible Errors During the FTP Network Access on the File System of the Memory Card	135
Table 55: Data Saved in the Directory Structure	136
Table 56: ETHERNET Libraries for WAGO-I/O-PRO.....	174
Table 57: Task Processing.....	180
Table 58: WBM Page “Information”	194
Table 59: WBM Page “Ethernet”	196
Table 60: WBM Page “Ethernet“	197
Table 61: WBM Page “TCP/IP”	199
Table 62: WBM page “Port”	200
Table 63: WBM page “SNMP”	204
Table 64: WBM Page “SNMP V3”	206
Table 65: WBM Page “Watchdog”	208
Table 66: WBM Page “Clock”	211
Table 67: WBM Page “Security”	214
Table 68: WBM Page “Modbus”	216
Table 69: WBM Page “BACnet”	217
Table 70: WBM Page “BACnet”	218
Table 71: WBM Page “PLC Info”	219
Table 72: WBM Page “PLC”	221
Table 73: WBM Page „PLC“	223
Table 74: WBM Page “Features”	226
Table 75: WBM Page “I/O Config”	228
Table 76: WBM Page “Disk Info”	230
Table 77: WBM Page “SD Card”	232
Table 78: WBM Page “Backup & Restore”	234
Table 79: LED Assignment for Diagnostics	238
Table 80: Fieldbus Diagnostics – Solution in Event of Error	239
Table 81: Node Status Diagnostics – Solution in Event of Error.....	240
Table 82: Blink Code Table for the I/O LED Signaling, Error Code 1	243
Table 83: Blink Code Table for the I/O LED Signaling, Error Code 2	244
Table 84: Blink Code Table for the I/O LED Signaling, Error Code 3	245
Table 85: Blink Code Table for the I/O LED Signaling, Error Code 4	246
Table 86: Blink Code Table for the I/O LED Signaling, Error Code 5	246
Table 87: Blink Code Table for the I/O LED Signaling, Error Code 6	247
Table 88: Blink Code Table for the I/O LED Signaling, Error Code 7	247
Table 89: Blink Code Table for the I/O LED Signaling, Error Code 8 ... 9	248
Table 90: Blink Code Table for the I/O LED Signaling, Error Code 10	248

Table 91: Blink Code Table for the I/O LED Signaling, Error Code 11	249
Table 92: Blink code table for I/O LED signaling, error code 12.....	249
Table 93: Blink Code Table for the I/O LED Signaling, Error Code 14	250
Table 94: Power Supply Status Diagnostics – Solution in Event of Error.....	251
Table 95: IP Packet.....	254
Table 96: Network Class A	255
Table 97: Network Class B	255
Table 98: Network Class C	255
Table 99: Key Data Class A, B and C.....	256
Table 100: Example: Class B Address with Field for Subnet IDs	257
Table 101: Subnet Mask for Class A Network.....	257
Table 102: Subnet Mask for Class B Network.....	257
Table 103: Subnet Mask for Class C Network.....	257
Table 104: Example for an IP Address from a Class B Network.....	257
Table 105: BootP Options	261
Table 106: Meaning of DHCP Options.....	263
Table 107: Meaning of the SNTP Parameters.....	264
Table 108: FTP Commands and Function.....	266
Table 109: MIB II Groups.....	268
Table 110: Standard Traps	268
Table 111: BACnet Objects	270
Table 112: Priority Levels.....	277
Table 113: IA/BIBB of the Standardized Device Profiles	285
Table 114: MODBUS/TCP Header.....	293
Table 115: Basic Data Types of MODBUS Protocol.....	294
Table 116: List of the MODBUS Functions in the Fieldbus Controller	294
Table 117: Exception Codes.....	297
Table 118: Request of Function Code FC1	298
Table 119: Response of Function Code FC1.....	298
Table 120: Assignment of Inputs	298
Table 121: Exception of Function Code FC1.....	299
Table 122: Request of Function Code FC2	300
Table 123: Response of Function Code FC2.....	300
Table 124: Assignment of Inputs	300
Table 125: Exception of Function Code FC2.....	301
Table 126: Request of Function Code FC3.....	302
Table 127: Response of Function Code FC3.....	302
Table 128: Exception of Function Code FC3.....	302
Table 129: Request of Function Code FC4.....	303
Table 130: Response of Function Code FC4.....	303
Table 131: Exception of Function Code FC4.....	303
Table 132: Request of Function Code FC5	304
Table 133: Response of Function Code FC5.....	304
Table 134: Exception of Function Code FC5.....	304
Table 135: Request of Function Code FC6	305
Table 136: Response of Function Code FC6.....	305
Table 137: Exception of Function Code FC6.....	305
Table 138: Request of Function code FC11	306
Table 139: Response of Function Code FC11.....	306
Table 140: Exception of Function Code FC 11.....	306

Table 141: Request of Function Code FC15	307
Table 142: Response of Function Code FC15.....	307
Table 143: Exception of Function Code FC15.....	308
Table 144: Request of Function Code FC16	309
Table 145: Response of Function Code FC16.....	309
Table 146: Exception of Function Code FC16.....	309
Table 147: Request of Function Code FC22	310
Table 148: Response of Function Code FC22.....	310
Table 149: Exception of Function Code FC22.....	310
Table 150: Request of Function Code FC23	311
Table 151: Response of Function Code FC23.....	311
Table 152: Exception of Function Code FC23.....	311
Table 153: Register Access Reading (with FC3, FC4 and FC23).....	313
Table 154: Register Access Writing (with FC6, FC16, FC22 and FC23).....	314
Table 155: Bit Access Reading (with FC1 and FC2).....	315
Table 156: Bit Access Writing (with FC5 and FC15).....	315
Table 157: MODBUS Registers.....	316
Table 158: MODBUS registers (Continuation).....	317
Table 159: Register Address 0x1000	318
Table 160: Register Address 0x1001	318
Table 161: Register Address 0x1002	319
Table 162: Register Address 0x1003	319
Table 163: Register Address 0x1004	319
Table 164: Register Address 0x1005	319
Table 165: Register Address 0x1006	320
Table 166: Register Address 0x1007	320
Table 167: Register Address 0x1008	320
Table 168: Register Address 0x1009	320
Table 169: Register Address 0x100A	320
Table 170: Starting Watchdog.....	321
Table 171: Register Address 0x100B.....	322
Table 172: Register Address 0x1020	323
Table 173: Register Address 0x1021	323
Table 174: Register Address 0x1022	324
Table 175: Register Address 0x1023	324
Table 176: Register Address 0x1024	324
Table 177: Register Address 0x1025	324
Table 178: Register Address 0x1028	324
Table 179: Register Address 0x1029	325
Table 180: Register Address 0x102A	325
Table 181: Register Address 0x102B.....	325
Table 182: Register Address 0x1030	325
Table 183: Register Address 0x1031	325
Table 184: Register Address 0x1035	326
Table 185: Register Address 0x1036	326
Table 186: Register Address 0x1037	326
Table 187: Register Address 0x1050	326
Table 188: Register Address 0x2030	327
Table 189: Register Address 0x2031	327
Table 190: Register Address 0x2032	327

Table 191: Register Address 0x2033	328
Table 192: Register Address 0x2040	328
Table 193: Register Address 0x2041	328
Table 194: Register Address 0x2042	328
Table 195: Register Address 0x2043	328
Table 196: Register Address 0x2010	329
Table 197: Register Address 0x2011	329
Table 198: Register Address 0x2012	329
Table 199: Register Address 0x2013	329
Table 200: Register Address 0x2014	329
Table 201: Register Address 0x2020	329
Table 202: Register Address 0x2021	330
Table 203: Register Address 0x2022	330
Table 204: Register Address 0x2023	330
Table 205: Register Address 0x2000	331
Table 206: Register Address 0x2001	331
Table 207: Register Address 0x2002	331
Table 208: Register Address 0x2003	331
Table 209: Register Address 0x2004	331
Table 210: Register Address 0x2005	331
Table 211: Register Address 0x2006	332
Table 212: Register Address 0x2007	332
Table 213: Register Address 0x2008	332
Table 214: Register Addresses 0x3000 to 0x5FFF	332
Table 215: 1 Channel Digital Input Module with Diagnostics	335
Table 216: 2 Channel Digital Input Modules.....	335
Table 217: 2 Channel Digital Input Module with Diagnostics	335
Table 218: 2 Channel Digital Input Module with Diagnostics and Output Process Data	336
Table 219: 4 Channel Digital Input Modules.....	336
Table 220: 8 Channel Digital Input Modules.....	336
Table 221: 8 Channel Digital Input Module PTC with Diagnostics and Output Process Data.....	337
Table 222: 16 Channel Digital Input Modules.....	337
Table 223: 1 Channel Digital Output Module with Input Process Data	338
Table 224: 2 Channel Digital Output Modules	338
Table 225: 2 Channel Digital Input Modules with Diagnostics and Input Process Data	339
Table 226: 2 Channel Digital Input Modules with Diagnostics and Input Process Data 75x-506.....	339
Table 227: 4 Channel Digital Output Modules	340
Table 228: 4 Channel Digital Output Modules with Diagnostics and Input Process Data	340
Table 229: 8 Channel Digital Output Module.....	340
Table 230: 8 Channel Digital Output Modules with Diagnostics and Input Process Data	341
Table 231: 16 Channel Digital Output Modules	341
Table 232: 8 Channel Digital Input/Output Modules.....	342
Table 233: 1 Channel Analog Input Modules	343
Table 234: 2 Channel Analog Input Modules	343

Table 235: 4 Channel Analog Input Modules	344
Table 236: 3-Phase Power Measurement Module.....	345
Table 237: 8 Channel Analog Input Modules	346
Table 238: 2 Channel Analog Output Modules.....	347
Table 239: 4 Channel Analog Output Modules.....	347
Table 240: 8 Channel Analog Output Modules.....	348
Table 241: Counter Modules 750-404, (and all variations except of /000-005), 753-404, (and variation /000-003)	349
Table 242: Counter Modules 750-404/000-005	350
Table 243: Counter Modules 750-638, 753-638	350
Table 244: Pulse Width Modules 750-511, /xxx-xxx	351
Table 245: Serial Interface Modules with alternative Data Format	351
Table 246: Serial Interface Modules with Standard Data Format.....	352
Table 247: Data Exchange Module	352
Table 248: SSI Transmitter Interface Modules	353
Table 249: Incremental Encoder Interface Modules 750-631/000-004, --010, -011	353
Table 250: Incremental Encoder Interface Modules 750-634.....	354
Table 251: Incremental Encoder Interface Modules 750-637.....	354
Table 252: Digital Pulse Interface Modules 750-635	355
Table 253: DC-Drive Controller 750-636	355
Table 254: Stepper Controller RS 422 / 24 V / 20 mA 750-670.....	356
Table 255: RTC Module 750-640	357
Table 256: DALI/DSI Master Module 750-641.....	357
Table 257: Overview of Input Process Image in the "Easy" Mode.....	359
Table 258: Overview of the Output Process Image in the "Easy" Mode“	359
Table 259: EnOcean Radio Receiver 750-642	360
Table 260: MP Bus Master Module 750-643.....	361
Table 261: Bluetooth® RF-Transceiver 750-644.....	361
Table 262: Vibration Velocity/Bearing Condition Monitoring VIB I/O 750-645	362
Table 263: KNX/EIB/TP1 Module 753-646	363
Table 264: AS-interface Master Module 750-655	364
Table 265: System Modules with Diagnostics 750-610, -611	365
Table 266: Binary Space Module 750-622 (with Behavior Like 2 Channel Digital Input).....	365
Table 267: MODBUS Table and Function Codes.....	368
Table 268: Description of Marking Example for Approved I/O Modules According to ATEX and IECEx	371
Table 269: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx	373
Table 270: Description of Marking Example for Approved I/O Modules According to NEC 500.....	375
Table 271: Interrelationships for the Polarity Property	405
Table 272: Interrelationships for the Polarity Property (Binary Input).....	406
Table 273: Interrelationships for BACnet Polarity (Binary Output).....	406
Table 274: MIB II – System Group.....	418
Table 275: MIB II – Interface Group	419
Table 276: MIB II – IP Group.....	421
Table 277: MIB II – IpRoute Table Group	422

Table 278: MIB II – ICMP Group.....	423
Table 279: MIB II – TCP Group	424
Table 280: MIB II – UDP Group	425
Table 281: MIB II – SNMP Group	426
Table 282: WAGO MIB – Company Group	427
Table 283: WAGO MIB – Product Group	427
Table 284: WAGO MIB – Versions Group	428
Table 285: WAGO MIB – Real Time Clock Group	429
Table 286: WAGO MIB – Ethernet Group	430
Table 287: WAGO MIB – Actual Error Group	430
Table 288: WAGO MIB – PLC Project Group	431
Table 289: WAGO MIB – Http Group	432
Table 290: WAGO MIB – Ftp Group	432
Table 291: WAGO MIB – Sntp Group	433
Table 292: WAGO MIB – Snmp Group	433
Table 293: WAGO MIB – Snmp Trap String Group.....	435
Table 294: WAGO MIB – Snmp User Trap String Group	436
Table 295: WAGO MIB – Plc Connection Group	436
Table 296: WAGO MIB – Modbus Group	437
Table 297: WAGO MIB – Process Image Group	438
Table 298: WAGO MIB – Plc Data Group	439

WE! INNOVATE!

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: 05 71/8 87 – 0
Fax: 05 71/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>

