WAGO-I/O-SYSTEM 750

Modular I/O System

BACnet/IP Controller

750-830

Manual

Technical description,
installation and
configuration

Version 1.0.1

WAGO
INNOVATIVE CONNECTIONS

The contents of this documentation are taken in part from the BACnet Standard 135-2004 or are based on the original contents. These contents are subject to copyright.

The following applies to these contents:

The following applies for the BACnet logo:

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 correctness and completeness of this documentation. However, as errors can never be fully excluded we would appreciate any information or ideas at any time.

E-Mail: documentation@wago.com

Content

1 Important Notes

This section provides only a summary of the most important safety requirements and notes which will be mentioned in the individual sections. To protect your health and prevent damage to the devices, it is essential to read and carefully follow the safety guidelines.

1.1 Legal Principles

1.1.1 Copyright

This manual including all figures and illustrations contained therein is subject to copyright. Any use of this manual which infringes the copyright provisions stipulated herein, is not permitted. Reproduction, translation and electronic and phototechnical archiving and amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden. Non-observance will entail the right of claims for damages.

WAGO Kontakttechnik GmbH & Co. KG reserves the right of changes serving technical progress.
All rights developing from the issue of a patent or the legal protection of utility patents are reserved to WAGO Kontakttechnik GmbH & Co. KG. Third-party products are always indicated without any notes concerning patent rights. Thus, the existence of such rights must not be excluded.

1.1.2 Personnel Qualification

The use of the product described in this manual requires special qualifications, as shown in the following table:

Activity	Electrical specialist	Instructed personnel*)	Specialists**) having qualifications in PLC programming
Assembly	X	X	
Commissioning	X		X
Programming			X
Maintenance	X	X	
Troubleshooting	X		
Disassembly	X	X	

*) Instructed persons have been trained by qualified personnel or electrical specialists.

**) A specialist is someone who, through technical training, knowledge and experience, demonstrates the ability to meet the relevant specifications and identify potential dangers in the mentioned field of activity.

All personnel must be familiar with the applicable standards.
WAGO Kontakttechnik GmbH & Co. KG declines any liability resulting from improper action and damage to WAGO products and third party products due to non-observance of the information contained in this manual.

1.1.3 Conforming Use of Series 750

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

The device is designed for IP20 protection class. It is protected against finger touch and solid impurities up to 12.5mm diameter, but not against water penetration. Unless otherwise specified, the device must not be operated in wet and dusty environments.

1.1.4 Technical Condition of the Devices

For each individual application, the components are supplied from the factory with a dedicated hardware and software configuration. Changes in hardware, software and firmware are only admitted within the framework of the possibilities documented in the manuals. All changes to the hardware or software and the non-conforming use of the components entail the exclusion of liability on the part of WAGO Kontakttechnik GmbH & Co. KG.
Please direct any requirements pertaining to a modified and/or new hardware or software configuration directly to WAGO Kontakttechnik GmbH & Co. KG.

1.2 Standards and Regulations for Operating the 750 Series

Please observe the standards and regulations that are relevant to your installation:

- The data and power lines must be connected and installed in compliance with the standards to avoid failures on your installation and eliminate any danger to personnel.
- For installation, startup, maintenance and repair, please observe the accident prevention regulations of your machine (e.g. BGV A 3, "Electrical Installations and Equipment").
- Emergency stop functions and equipment must not be made ineffective. See relevant standards (e.g. DIN EN 418).
- Your installation must be equipped in accordance to the EMC guidelines so that electromagnetic interferences can be eliminated.
- Operating 750 Series components in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section on "WAGO-I/O-SYSTEM 750" → "System Description" → "Technical Data".
- Please observe the safety measures against electrostatic discharge according to DIN EN 61340-5-1/-3. When handling the modules, ensure that the environment (persons, workplace and packing) is well grounded.
- The relevant valid and applicable standards and guidelines concerning the installation of switch cabinets are to be observed.

1.3 Symbols

Danger
Always observe this information to protect persons from injury.

Warning
Always observe this information to prevent damage to the device.

Attention
Marginal conditions that must always be observed to ensure smooth and efficient operation.

ESD (Electrostatic Discharge)
Warning of damage to the components through electrostatic discharge. Observe the precautionary measure for handling components at risk of electrostatic discharge.

Note
Make important notes that are to be complied with so that a trouble-free and efficient device operation can be guaranteed.

Additional Information
References to additional literature, manuals, data sheets and internet pages.

1.4 Safety Information

When connecting the device to your installation and during operation, the following safety notes must be observed:

Danger
The WAGO-I/O-SYSTEM 750 and its components are an open system. It must only be assembled in housings, cabinets or in electrical operation rooms. Access is only permitted via a key or tool to authorized qualified personnel.

Danger
All power sources to the device must always be switched off before carrying out any installation, repair or maintenance work.

Warning
Replace defective or damaged device/module (e.g. in the event of deformed contacts), as the functionality of field bus station in question can no longer be ensured on a long-term basis.

Warning
The components are not resistant against materials having seeping and insulating properties. Belonging to this group of materials is: e.g. aerosols, silicones, triglycerides (found in some hand creams). If it cannot be ruled out that these materials appear in the component environment, then the components must be installed in an enclosure that is resistant against the above mentioned materials. Clean tools and materials are generally required to operate the device/module.

Warning
Soiled contacts must be cleaned using oil-free compressed air or with ethyl alcohol and leather cloths.

Warning
Do not use contact sprays, which could possibly impair the functioning of the contact area.

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

ESD (Electrostatic Discharge)
The devices are equipped with electronic components that may be destroyed by electrostatic discharge when touched.

Warning
For components with ETHERNET/RJ-45 connectors:
Only for use in LAN, not for connection to telecommunication circuits.

1.5 Font Conventions

italic	Names of paths and files are marked in italic. e.g.: *C:\Programs\WAGO-IO-CHECK*
italic	Menu items are marked in bold italic. e.g.: ***Save***
\	A backslash between two names characterizes the selection of a menu point from a menu. e.g.: ***File \ New***
END	Press buttons are marked as bold with small capitals e.g.: **ENTER**
< >	Keys are marked bold within angle brackets e.g.: **<F5>**
`Courier`	The print font for program codes is Courier. e.g.: `END_VAR`

1.6 Number Notation

Number code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	Within inverted commas, Nibble separated with dots

1.7 Scope

This manual describes the field bus independent WAGO I/O SYSTEM 750 with the programmable **BACnet/IP Controller**.

Item.-No.	Description
750-830	**BACnet/IP Controller**

1.8 Abbreviation

AI	Analog Input
AO	Analog Output
DI	Digital Input
DO	Digital Output
I/O	Input/Output
ID	Identifier
PFC	Programmable Fieldbus Controller

2 The WAGO-I/O-SYSTEM 750

2.1 System Description

The WAGO-I/O-SYSTEM 750 is a modular, field bus independent I/O system. It is comprised of a field bus coupler/controller (1) and connected field bus modules (2) for any type of signal. Together, these make up the field bus node. The end module (3) completes the node.

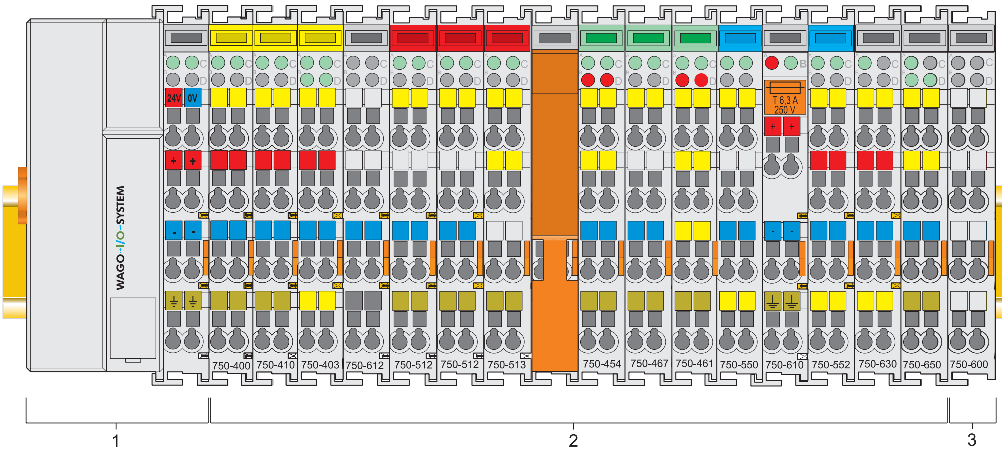


Fig. 2-1: Field bus node g0xxx00x

Couplers/controllers for field bus systems such as PROFIBUS, INTERBUS, ETHERNET TCP/IP, CAN (CANopen, DeviceNet, CAL), MODBUS, LON and others are available.

The coupler/controller contains the field bus interface, electronics and a power supply terminal. The field bus interface forms the physical interface to the relevant field bus. The electronics process the data of the bus modules and make it available for the field bus communication. The 24 V system supply and the 24 V field supply are fed in via the integrated power supply terminal. The field bus coupler communicates via the relevant field bus. The programmable field bus controller (PFC) enables the implementation of additional PLC functions. Programming is done with the WAGO I/O *PRO* CAA in accordance with IEC 61131-3.

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

The WAGO-I/O-SYSTEM 750 has a clear port level with LEDs for status indication, insertable mini WSB markers and pullout group marker carriers. The 3-wire technology supplemented by a ground wire connection allows for direct sensor/actuator wiring.

2.2 Technical Data

Mechanic	
Material	Polycarbonate, Polyamide 6.6
Dimensions W x H* x L * from upper edge of DIN 35 rail - Coupler/Controller (Standard) - Coupler/Controller (ECO) - Coupler/Controller (FireWire) - I/O module, single - I/O module, double - I/O module, fourfold	 - 51 mm x 65 mm x 100 mm - 50 mm x 65 mm x 100 mm - 62 mm x 65 mm x 100 mm - 12 mm x 64 mm x 100 mm - 24 mm x 64 mm x 100 mm - 48 mm x 64 mm x 100 mm
Installation	on DIN 35 with interlock
Modular by	double feather key dovetail
Mounting position	any position
Marking	standard marking label type group marking label 8 x 47 mm
Connection	
Connection type	CAGE CLAMP®
Wire range	0.08 mm² ... 2.5 mm², AWG 28-14
Stripped length	8 … 9 mm, 9 … 10 mm for components with pluggable wiring (753-xxx)
Contacts	
Power jumpers contacts	blade/spring contact self-cleaning
Current via power contacts $_{max}$	10 A
Voltage drop at I_{max}	< 1 V/64 modules
Data contacts	slide contact, hard gold plated 1.5 µm, self-cleaning
Climatic environmental conditions	
Operating temperature	0 °C ... 55 °C, -20 °C … +60 °C for components with extended temperature range (750-xxx/025-xxx)
Storage temperature	-20 °C ... +85 °C
Relative humidity	5 % … 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%	$SO_2 \leq 25$ ppm $H_2S \leq 10$ ppm
Special conditions	Ensure that additional measures for components are taken, which are used in an environment involving: – dust, caustic vapors or gases – ionization radiation

Safe electrical isolation				
Air and creepage distance	acc. to IEC 60664-1			
Degree of pollution acc. To IEC 61131-2	2			
Degree of protection				
Degree of protection	IP 20			
Electromagnetic compatibility				
Immunity to interference for industrial areas acc. to EN 61000-6-2 (2001)				
Test specification	**Test values**		**Strength class**	**Evaluation criteria**
EN 61000-4-2 ESD	4 kV/8 kV (contact/air)		2/3	B
EN 61000-4-3 electromagnetic fields	10 V/m 80 MHz ... 1 GHz		3	A
EN 61000-4-4 burst	1 kV/2 kV (data/supply)		2/3	B
EN 61000-4-5 surge	Data:	-/- (line/line)		B
		1 kV (line/earth)	2	
	DC supply:	0.5 kV (line/line)	1	B
		0.5 kV (line/earth)	1	
	AC supply:	1 kV (line/line)	2	B
		2 kV (line/earth)	3	
EN 61000-4-6 RF disturbances	10 V/m 80 % AM (0.15 ... 80 MHz)		3	A

Emission of interference for industrial areas acc. to EN 61000-6-4 (2001)			
Test specification	**Limit values/[QP]*)**	**Frequency range**	**Distance**
EN 55011 (AC supply, conducted)	79 dB (µV)	150 kHz ... 500 kHz	
	73 dB (µV)	500 kHz ... 30 MHz	
EN 55011 (radiated)	40 dB (µV/m)	30 MHz ... 230 MHz	10 m
	47 dB (µV/m)	230 MHz ... 1 GHz	10 m
Emission of interference for residential areas acc. to EN 61000-6-3 (2001)			
Test specification	**Limit values/[QP]*)**	**Frequency range**	**Distance**
EN 55022 (AC supply, conducted)	66 ... 56 dB (µV)	150 kHz ... 500 kHz	
	56 dB (µV)	500 kHz ... 5 MHz	
	60 dB (µV)	5 MHz ... 30 MHz	
EN 55022 (DC supply/data, conducted)	40 ... 30 dB (µA)	150 kHz ... 500 kHz	
	30 dB (µA)	500 kHz ... 30 MHz	
EN 55022 (radiated)	30 dB (µV/m)	30 MHz ... 230 MHz	10 m
	37 dB (µV/m)	230 MHz ... 1 GHz	10 m

Mechanical strength acc. to IEC 61131-2		
Test specification	**Frequency range**	**Limit value**
IEC 60068-2-6 vibration	5 Hz ≤ f < 9 Hz	1.75 mm amplitude (permanent) 3.5 mm amplitude (short term)
	9 Hz ≤ f < 150 Hz	0.5 g (permanent) 1 g (short term)
	Note on vibration test: a) Frequency change: max. 1 octave/minute b) Vibration direction: 3 axes	
IEC 60068-2-27 shock		15 g
	Note on shock test: a) Type of shock: half sine b) Shock duration: 11 ms c) Shock direction: 3x in positive and 3x in negative direction for each of the three mutually perpendicular axes of the test specimen	
IEC 60068-2-32 free fall		1 m (module in original packing)

*) QP: Quasi Peak

Note
If the technical data of components differ from the values described here, the technical data shown in the manuals of the respective components shall be valid.

For Products of the WAGO-I/O-SYSTEM 750 with ship specific approvals supplementary guidelines are valid:

Electromagnetic compatibility			
Immunity to interference acc. to Germanischer Lloyd (2003)			
Test specification	**Test values**	**Strength class**	**Evaluation criteria**
IEC 61000-4-2 ESD	6 kV/8 kV (contact/air)	3/3	B
IEC 61000-4-3 electromagnetic fields	10 V/m 80 MHz ... 2 GHz	3	A
IEC 61000-4-4 burst	1 kV/2 kV (data/supply)	2/3	A
IEC 61000-4-5 surge	AC/DC Supply: 0.5 kV (line/line)	1	A
	AC/DC Supply: 1 kV (line/earth)	2	
IEC 61000-4-6 RF disturbances	10 V/m 80 % AM (0.15 ... 80 MHz)	3	A
Type test AF disturbances (harmonic waves)	3 V, 2 W	-	A
Type test high voltage	755 V DC 1500 V AC	-	-

Emission of interference acc. to Germanischer Lloyd (2003)			
Test specification	**Limit values**	**Frequency range**	**Distance**
Type test (EMC1, conducted) allows for ship bridge control applications	96 ... 50 dB (µV)	10 kHz ... 150 kHz	
	60 ... 50 dB (µV)	150 kHz ... 350 kHz	
	50 dB (µV)	350 kHz ... 30 MHz	
Type test (EMC1, radiated) allows for ship bridge control applications	80 ... 52 dB (µV/m)	150 kHz ... 300 kHz	3 m
	52 ... 34 dB (µV/m)	300 kHz ... 30 MHz	3 m
	54 dB (µV/m)	30 MHz ... 2 GHz	3 m
except:	24 dB (µV/m)	156 MHz ... 165 MHz	3 m

Mechanical strength acc. to Germanischer Lloyd (2003)		
Test specification	**Frequency range**	**Limit value**
IEC 60068-2-6 vibration (category A – D)	2 Hz ≤ f < 25 Hz	± 1.6 mm amplitude (permanent)
	25 Hz ≤ f < 100 Hz	4 g (permanent)
	Note on vibration test: a) Frequency change: max. 1 octave/minute b) Vibration direction: 3 axes	

Range of application	Required specification emission of interference	Required specification immunity to interference
Industrial areas	EN 61000-6-4 (2001)	EN 61000-6-2 (2001)
Residential areas	EN 61000-6-3 (2001)*)	EN 61000-6-1 (2001)

*) The system meets the requirements on emission of interference in residential areas with the field bus coupler/controller for:

ETHERNET	750-342/-841/-842/-860
LonWorks	750-319/-819
CANopen	750-337/-837
DeviceNet	750-306/-806
MODBUS	750-312/-314/ -315/ -316 750-812/-814/ -815/ -816
KNX	750-849
BACnet	750-830

With a special permit, the system can also be implemented with other field bus couplers/controllers in residential areas (housing, commercial and business areas, small-scale enterprises). The special permit can be obtained from an authority or inspection office. In Germany, the Federal Office for Post and Telecommunications and its branch offices issues the permit.

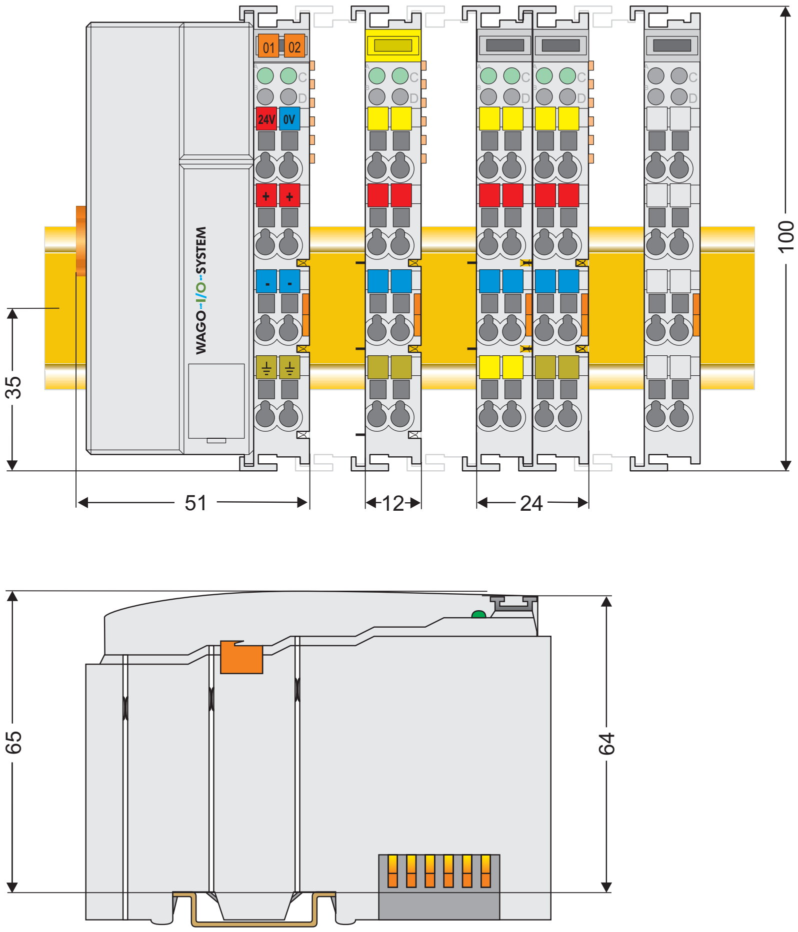
It is possible to use other field bus couplers/controllers under certain boundary conditions. Please contact WAGO Kontakttechnik GmbH & Co. KG.

Maximum power dissipation of the components	
Bus modules	0.8 W / bus terminal (total power dissipation, system/field)
Field bus coupler/controller	2.0 W / coupler/controller

Warning
The power dissipation of all installed components must not exceed the maximum conductible power of the housing (cabinet).

When dimensioning the housing, care is to be taken that even under high external temperatures, the temperature inside the housing does not exceed the permissible ambient temperature of 55 °C.

Dimensions



Side view

Dimensions in mm

Fig. 2-2: Dimensions g01xx05e

Note
The illustration shows a standard coupler. For detailed dimensions, please refer to the technical data of the respective coupler/controller.

2.3 Manufacturing Number

The manufacturing number indicates the delivery status directly after production.
This number is part of the lateral marking on the component.
In addition, starting from calendar week 43/2000 the manufacturing number is also printed on the cover of the configuration and programming interface of the field bus coupler or controller.

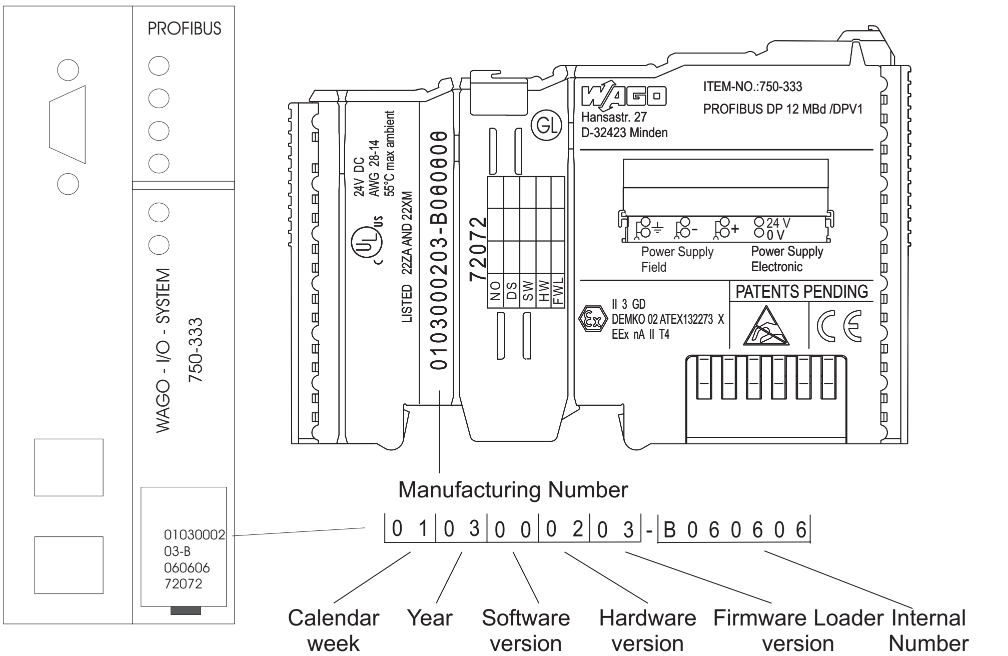


Fig. 2-3: Example: Manufacturing Number of a PROFIBUS field bus coupler 750-333

g01xx15e

The manufacturing number consists of the production week and year, the software version (if available), the hardware version of the component, the firmware loader (if available) and further internal information for WAGO Kontakttechnik GmbH & Co. KG.

2.4 Component Update

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

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

Update Matrix

Current Version data for:		1. Update	2. Update	3. Update	
Production Order Number	**NO**				← only starting from calendar week 13/2004
Datestamp	**DS**				
Software index	**SW**				
Hardware index	**HW**				
Firmware loader index	**FWL**				← only for coupler/controller

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

Additionally with the update of a field bus coupler or controller also the cover of the configuration and programming interface of the coupler or controller is printed on with the current manufacturing and production order number.

The original manufacturing data on the housing of the component remain thereby.

2.5 Storage, Assembly and Transport

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

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

Statically shielded transport bags with metal coatings are to be used for the transport of open components for which soiling with amine, amide and silicone has been ruled out, e.g. 3M 1900E.

2.6 Mechanical Setup

2.6.1 Installation Position

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

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

WAGO item 249-116	End stop for DIN 35 rail, 6 mm wide
WAGO item 249-117	End stop for DIN 35 rail, 10 mm wide

2.6.2 Total Expansion

The length of the module assembly (including one end module of 12mm width) that can be connected to the coupler/controller is 780 mm. When assembled, the I/O modules have a maximum length of 768 mm.

Examples:

- 64 I/O modules of 12 mm width can be connected to one coupler/controller.
- 32 I/O modules of 24 mm width can be connected to one coupler/controller.

Exception:

The number of connected I/O modules also depends on which type of coupler/controller is used. For example, the maximum number of I/O modules that can be connected to a PROFIBUS coupler/controller is 63 without end module. The maximum total expansion of a node is calculated as follows:

Warning
The maximum total length of a node without coupler/controller must not exceed 780 mm. Furthermore, restrictions made on certain types of couplers/controllers must be observed (e.g. for PROFIBUS).

2.6.3 Assembly onto Carrier Rail

2.6.3.1 Carrier Rail Properties

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

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

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

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

2.6.3.2 WAGO DIN Rail

WAGO carrier rails meet the electrical and mechanical requirements.

Item Number	Description
210-113 /-112	35 x 7.5; 1 mm; steel yellow chromated; slotted/unslotted
210-114 /-197	35 x 15; 1.5 mm; steel yellow chromated; slotted/unslotted
210-118	35 x 15; 2.3 mm; steel yellow chromated; unslotted
210-198	35 x 15; 2.3 mm; copper; unslotted
210-196	35 x 7.5; 1 mm; aluminum; unslotted

2.6.4 Spacing

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

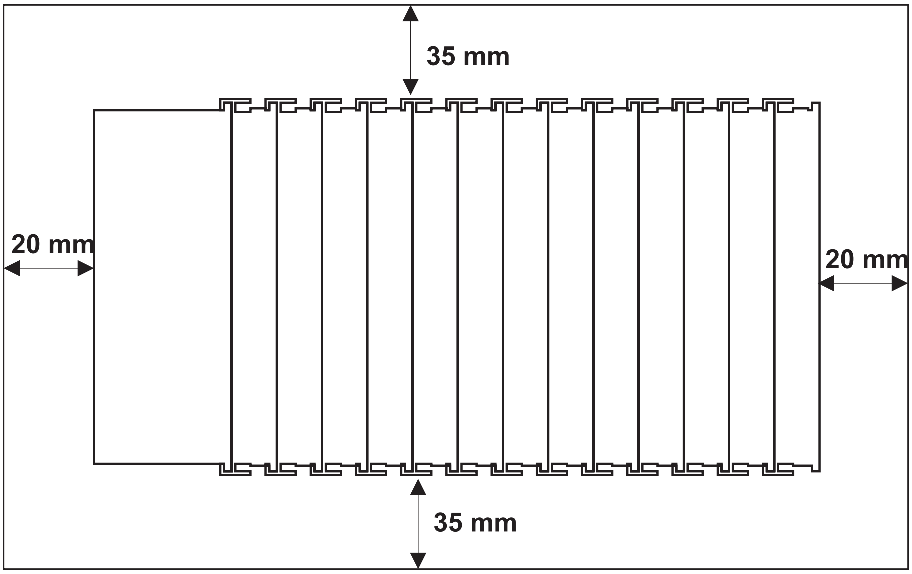


Fig. 2-4: Spacing g01xx13x

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

2.6.5 Plugging and Removal of the Components

Warning
Before work is done on the components, the voltage supply must be turned off.

In order to safeguard the coupler/controller from jamming, it should be fixed onto the carrier rail with the locking disc To do so, push on the upper groove of the locking disc using a screwdriver.

To pull out the field bus coupler/controller, release the locking disc by pressing on the bottom groove with a screwdriver and then pulling the orange colored unlocking lug .

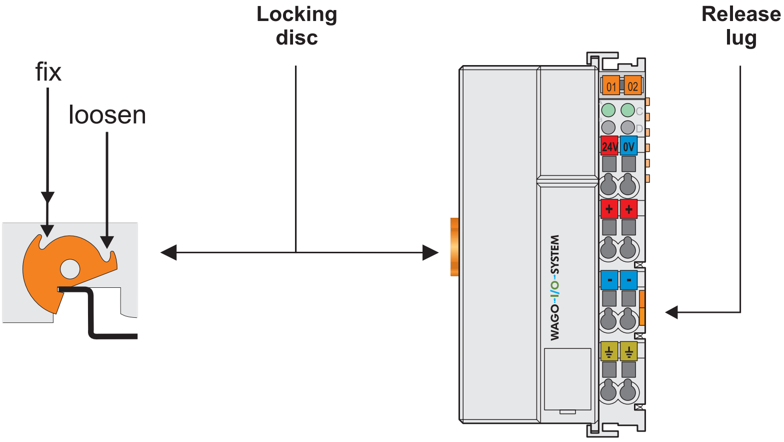


Fig. 2-5: Coupler/Controller and unlocking lug g01xx12e

It is also possible to release an individual I/O module from the unit by pulling an unlocking lug.

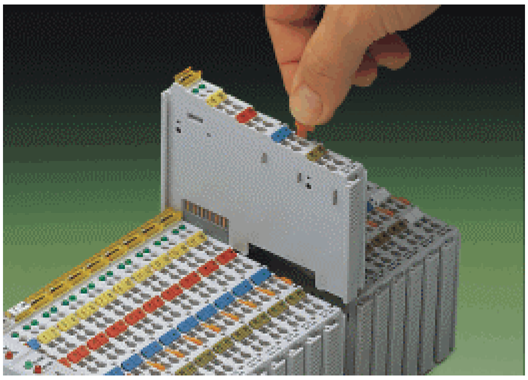


Fig. 2-6: removing bus terminal p0xxx01x

Danger
Ensure that an interruption of the PE will not result in a condition which could endanger a person or equipment!
For planning the ring feeding of the ground wire, please see chapter 2.6.3.

2.6.6 Assembly Sequence

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

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

Starting with the coupler/controller, the bus modules are assembled adjacent to each other according to the project planning. Errors in the planning of the node in terms of the potential groups (connection via the power contacts) are recognized, as the bus modules with power contacts (male contacts) cannot be linked to bus modules with fewer power contacts.

Attention
Always link the bus modules with the coupler/controller, and always plug from above.

Warning
Never plug bus modules from the direction of the end terminal. A ground wire power contact, which is inserted into a terminal without contacts, e.g. a 4-channel digital input module, has a decreased air and creepage distance to the neighboring contact in the example DI4.

Always terminate the field bus node with an end module (750-600).

2.6.7 Internal Bus/Data Contacts

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

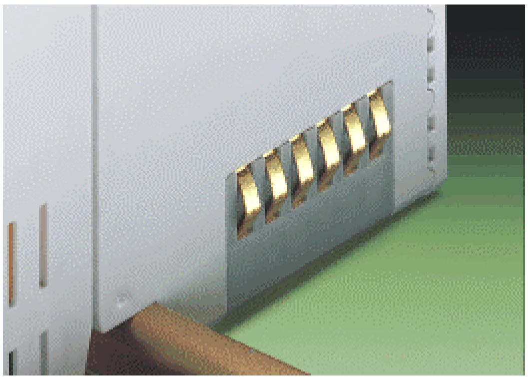


Fig. 2-7: Data contacts p0xxx07x

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

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

2.6.8 Power Contacts

Self-cleaning power contacts , are situated on the side of the components which further conduct the supply voltage for the field side. These contacts come as touchproof spring contacts on the right side of the coupler/controller and the bus module. As fitting counterparts the module has male contacts on the left side.

Danger
The male contacts are sharp-edged. Handle the module carefully to prevent injury.

Attention
Please take into consideration that some bus modules have no or only a few power jumper contacts. The design of some modules does not allow them to be physically assembled in rows, as the grooves for the male contacts are closed at the top.

Power jumper contacts

Blade	0	0	3	2
Spring	0	3	3	2

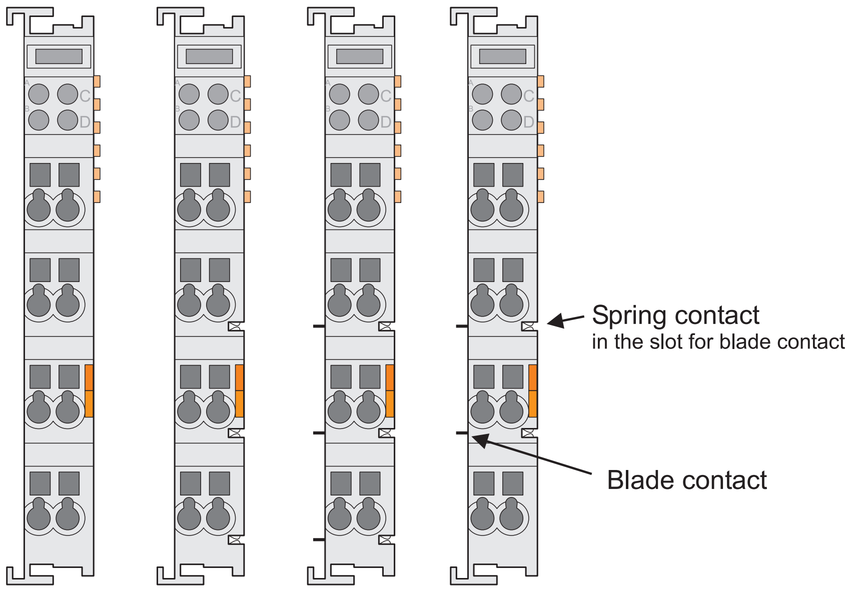


Fig. 2-8: Example for the arrangement of power contacts g0xxx05e

Recommendation
With the WAGO ProServe® Software smartDESIGNER, the structure of a field bus node can be configured. The configuration can be tested via the integrated accuracy check.

2.6.9 Wire Connection

All components have CAGE CLAMP® connections.

The WAGO CAGE CLAMP® connection is appropriate for solid, stranded and finely stranded conductors. Each clamping unit accommodates one conductor.

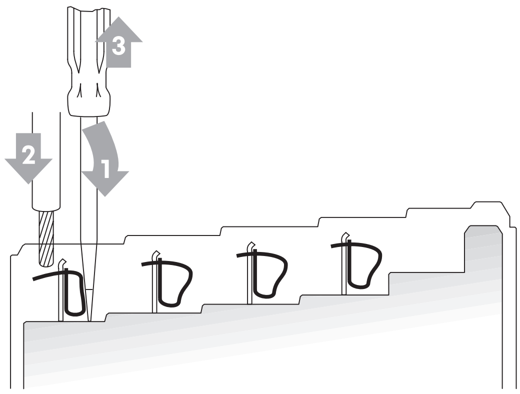


Fig. 2-9: CAGE CLAMP® Connection g0xxx08x

The operating tool is inserted into the opening above the connection. This opens the CAGE CLAMP®. Subsequently the conductor can be inserted into the opening. After removing the operating tool, the conductor is safely clamped.

More than one conductor per connection is not permissible. If several conductors have to be made at one connection point, then they should be made away from the connection point using WAGO Terminal Blocks. The terminal blocks may be jumpered together and a single wire brought back to the I/O module connection point.

Attention
If it is unavoidable to jointly connect 2 conductors, then a ferrule must be used to join the wires together.
Ferrule:

Length	8 mm
Nominal cross section $_{max.}$	1 mm^2 for 2 conductors with 0.5 mm^2 each
WAGO Product	216-103 or products with comparable properties

2.7 Power Supply

2.7.1 Isolation

Within the field bus node, there are three electrically isolated potentials.

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

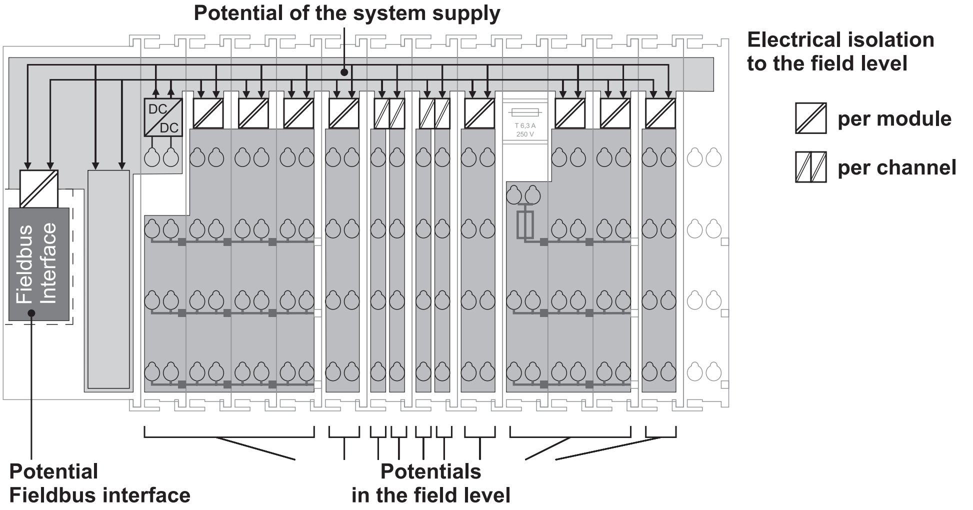


Fig. 2-10: Isolation g0xxx01e

Attention
The ground wire connection must be present in each group. In order that all protective conductor functions are maintained under all circumstances, it is recommended that a ground wire be connected at the beginning and end of a potential group. (ring format, please see chapter 2.8.3). Thus, if a bus module comes loose from a composite during servicing, then the protective conductor connection is still guaranteed for all connected field devices.

When using a joint power supply unit for the 24 V system supply and the 24 V field supply, the electrical isolation between the internal bus and the field level is eliminated for the potential group.

2.7.2 System Supply

2.7.2.1 Connection

The WAGO-I/O-SYSTEM 750 requires a 24 V direct current system supply (-15 % or +20 %). The power supply is provided via the coupler/controller and, if necessary, in addition via the internal system supply modules (750-613). The voltage supply is reverse voltage protected.

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

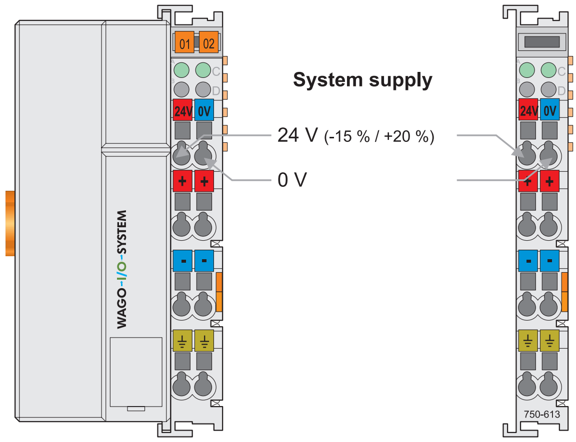


Fig. 2-11: System Supply g0xxx02e

The direct current supplies all internal system components, e.g. coupler/controller electronics, field bus interface and bus modules via the internal bus (5 V system voltage). The 5 V system voltage is electrically connected to the 24 V system supply.

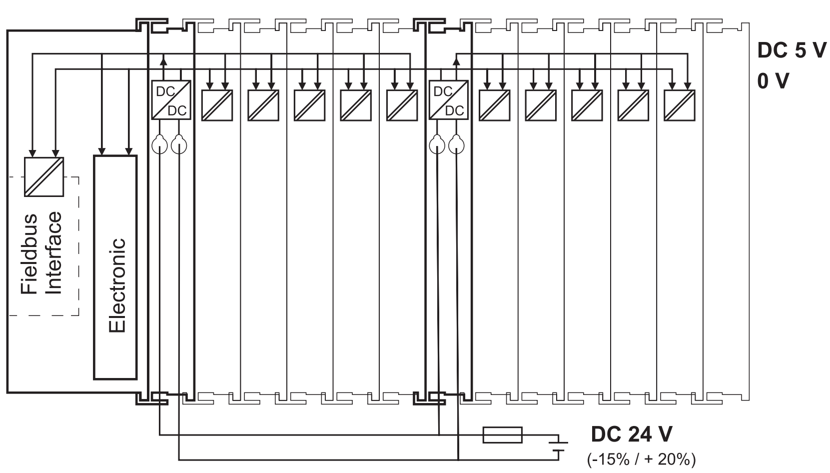


Fig. 2-12: System Voltage g0xxx06e

Attention
Resetting the system by switching on and off the system supply, must take place simultaneously for all supply modules (coupler/controller and 750-613).

2.7.2.2 Alignment

Recommendation
A stable network supply cannot be taken for granted always and everywhere. Therefore, regulated power supply units should be used in order to guarantee the quality of the supply voltage.

The supply capacity of the coupler/controller or the internal system supply module (750-613) can be taken from the technical data of the components.

Internal current consumption[*)]	Current consumption via system voltage: 5 V for electronics of the bus modules and coupler/controller
Residual current for bus terminals[*)]	Available current for the bus modules. Provided by the bus power supply unit. See coupler/controller and internal system supply module (750-613)

[*)] cf. catalogue W4 Volume 3, manuals or internet

Example	**Coupler 750-301:**	
	internal current consumption:	350 mA at 5 V
	residual current for bus modules:	1650 mA at 5 V
	sum $I_{(5V)\ total}$:	2000 mA at 5 V

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

Attention
If the *sum of the internal current consumption* exceeds the *residual current for bus modules*, then an internal system supply module (750-613) must be placed before the module where the permissible residual current was exceeded.

Example:

A node with a PROFIBUS Coupler 750-333 consists of 20 relay modules (750-517) and 10 digital input modules (750-405).

Current consumption:

20* 90 mA =	1800 mA	
10* 2 mA =		20 mA
Sum	1820 mA	

The coupler can provide 1650 mA for the bus modules. Consequently, an internal system supply module (750-613), e.g. in the middle of the node, should be added.

Recommendation
With the WAGO ProServe® Software smartDESIGNER, the assembly of a field bus node can be configured. The configuration can be tested via the integrated accuracy check.

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

Coupler/Controller

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

750-613

$I_{(5\ V)\ total}$ = *Sum of all the internal current consumption of the connected bus modules*

Input current $I_{(24\ V)}$ = $5\ V / 24\ V * I_{(5\ V)\ total} / \eta$

$\eta = 0.87$ *(at nominal load)*

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

During the test, all outputs, in particular those of the relay modules, must be active.

2.7.3 Field Supply

2.7.3.1 Connection

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

The coupler/controller provides field side power (DC 24V). In this case it is a passive power supply without protection equipment.
Power supply modules are available for other potentials, e. g. AC 230 V. Likewise, with the aid of the power supply modules, various potentials can be set up. The connections are linked in pairs with a power contact.

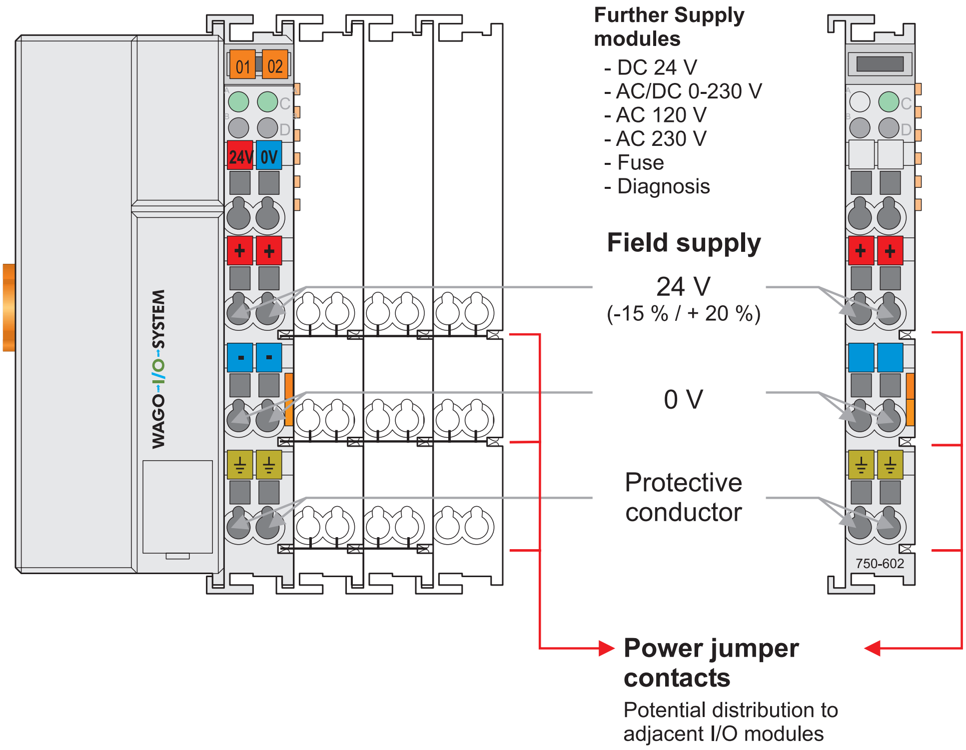


Fig. 2-13: Field Supply (Sensor/Actuator) g0xxx03e

The supply voltage for the field side is automatically passed to the next module via the power jumper contacts when assembling the bus modules .

The current load of the power contacts must not exceed 10 A on a continual basis. The current load capacity between two connection terminals is identical to the load capacity of the connection wires.

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

Attention
Some bus modules have no or very few power contacts (depending on the I/O function). Due to this, the passing through of the relevant potential is disrupted. If a field supply is required for subsequent bus modules, then a power supply module must be used.
Note the data sheets of the bus modules.

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

2.7.3.2 Fusing

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

750-601	24 V DC, Supply/Fuse
750-609	230 V AC, Supply/Fuse
750-615	120 V AC, Supply/Fuse
750-610	24 V DC, Supply/Fuse/Diagnosis
750-611	230 V AC, Supply/Fuse/Diagnosis

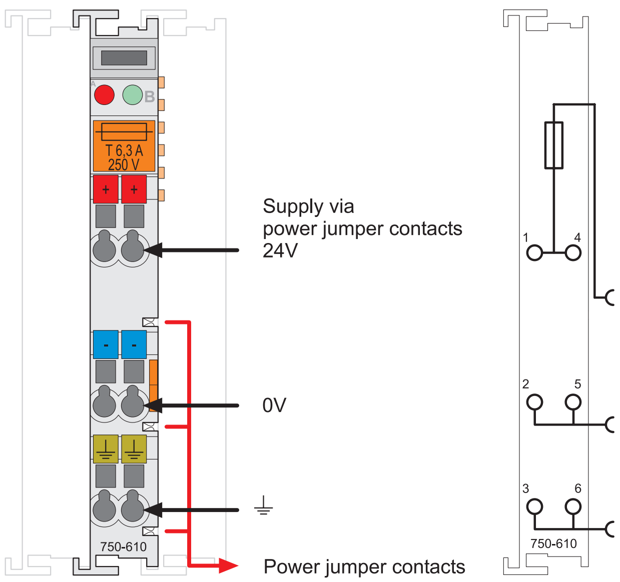


Fig. 2-14: Supply module with fuse carrier (Example 750-610) g0xxx09x

Warning
In the case of power supply modules with fuse holders, only fuses with a maximum dissipation of 1.6 W (IEC 127) must be used.
For UL approved systems only use UL approved fuses.

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

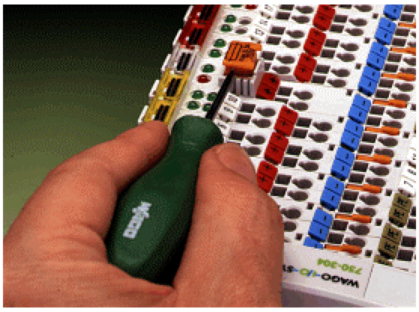


Fig. 2-15: Removing the fuse carrier p0xxx05x

Lifting the cover to the side opens the fuse carrier.

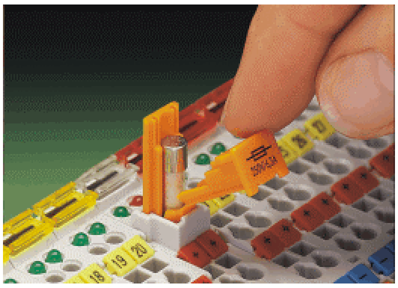


Fig. 2-16: Opening the fuse carrier p0xxx03x

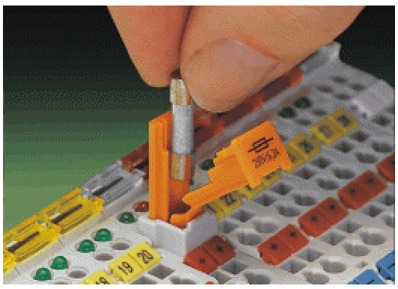


Fig. 2-17: Change fuse p0xxx04x

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

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

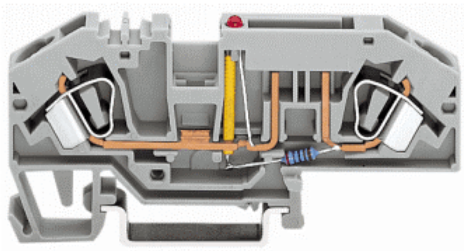


Fig. 2-18: Fuse modules for automotive fuses, series 282 pf66800x

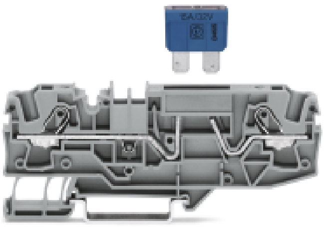


Abb. 2-19: Fuse modules for automotive fuses, series 2006 p0xxx13x

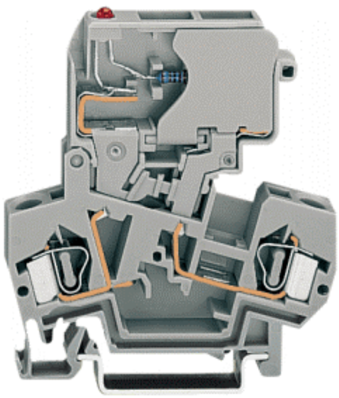


Fig. 2-20: Fuse modules with pivotable fuse carrier, series 281 pe61100x

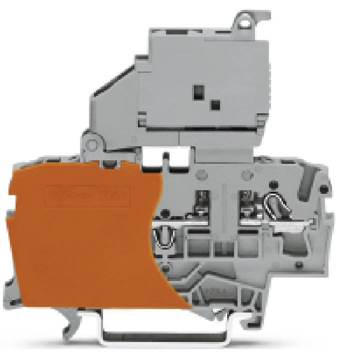


Abb. 2-21: Fuse modules with pivotable fuse carrier, series 2002 p0xxx12x

2.7.4 Supplementary Power Supply Regulations

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

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

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

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

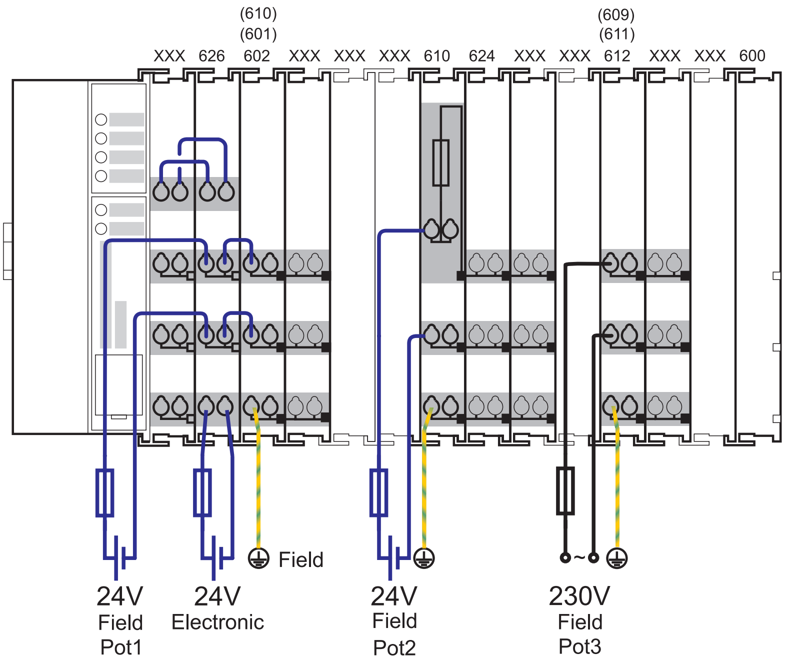


Fig. 2-22: Power supply concept g01xx11e

Note
Another potential power terminal 750-601/602/610 must only be used behind the filter terminal 750-626 if the protective earth conductor is needed on the lower power contact or if a fuse protection is required.

2.7.5 Supply Example

Attention
The system supply and the field supply should be separated in order to ensure bus operation in the event of a short-circuit on the actuator side.

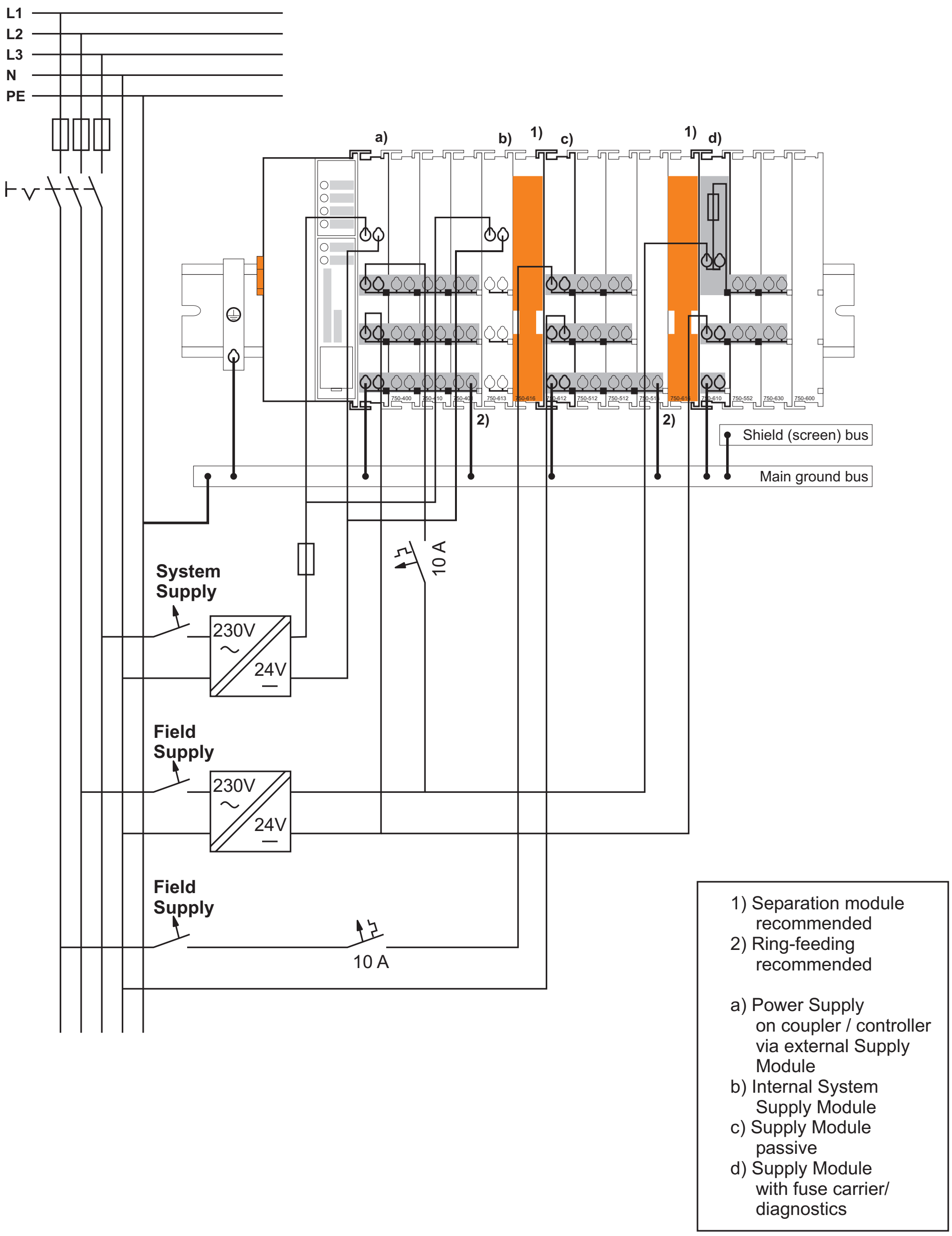


Fig. 2-23: Supply example g0xxx04e

2.7.6 Power Supply Unit

The WAGO-I/O-SYSTEM 750 requires a 24 V direct current system supply with a maximum deviation of -15 % or +20 %.

Recommendation
A stable network supply cannot be taken for granted always and everywhere. Therefore, regulated power supply units should be used in order to guarantee the quality of the supply voltage.

A buffer (200 µF per 1 A current load) should be provided for brief voltage dips. The I/O system buffers for approx 1 ms.

The electrical requirement for the field supply is to be determined individually for each power supply point. Thereby all loads through the field devices and bus modules should be considered. The field supply as well influences the bus modules, as the inputs and outputs of some bus modules require the voltage of the field supply.

Attention
The system supply and the field supply should be isolated from the power supplies in order to ensure bus operation in the event of short circuits on the actuator side.

WAGO products Item No.	Description
787-612	Primary switched mode; DC 24 V; 2,5 A Input nominal voltage AC 230 V
787-622	Primary switched mode; DC 24 V; 5 A Input nominal voltage AC 230 V
787-632	Primary switched mode; DC 24 V; 10 A Input nominal voltage AC 230/115 V
 288-809 288-810 288-812 288-813	Rail-mounted modules with universal mounting carrier AC 115 V / DC 24 V; 0,5 A AC 230 V / DC 24 V; 0,5 A AC 230 V / DC 24 V; 2 A AC 115 V / DC 24 V; 2 A

2.8 Grounding

2.8.1 Grounding the DIN Rail

2.8.1.1 Framework Assembly

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

Attention
Care must be taken to ensure the flawless electrical connection between the carrier rail and the frame or housing in order to guarantee sufficient grounding.

2.8.1.2 Insulated Assembly

Insulated assembly has been achieved when there is constructively no direct conduction connection between the cabinet frame or machine parts and the carrier rail. Here the earth must be set up via an electrical conductor.

The connected grounding conductor should have a cross section of at least 4 mm^2.

Recommendation
The optimal insulated setup is a metallic assembly plate with grounding connection with an electrical conductive link with the carrier rail.

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

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

2.8.2 Grounding Function

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

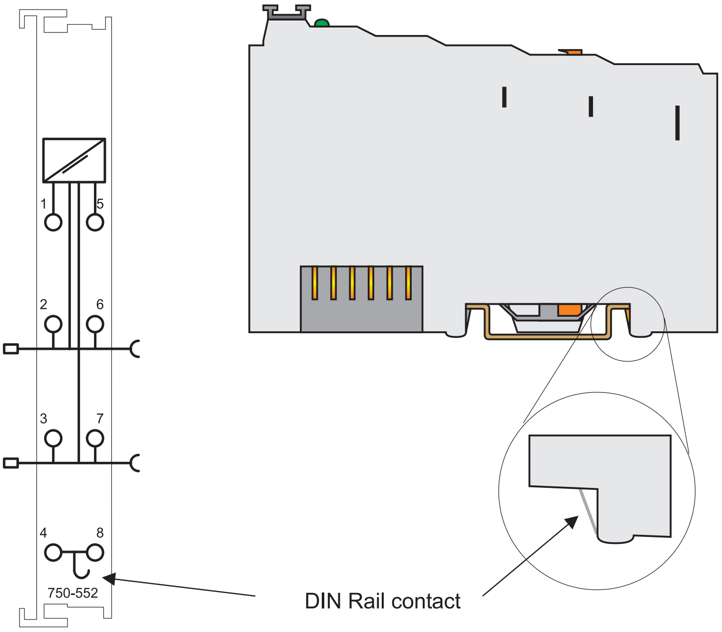


Fig. 2-24: Carrier rail contact g0xxx10e

Attention
Care must be taken to ensure the direct electrical connection between the carrier rail contact and the carrier rail.

The carrier rail must be grounded.

For information on carrier rail properties, please see chapter 2.6.3.2.

2.8.3 Grounding Protection

For the field side, the ground wire is connected to the lowest connection terminals of the power supply module. The ground connection is then connected to the next module via the Power Jumper Contact (PJC). If the bus module has the lower power jumper contact, then the ground wire connection of the field devices can be directly connected to the lower connection terminals of the bus module.

Attention
Should the ground conductor connection of the power jumper contacts within the node become disrupted, e. g. due to a 4-channel bus terminal, the ground connection will need to be re-established.

The ring feeding of the grounding potential will increase the system safety. When one bus module is removed from the group, the grounding connection will remain intact.

The ring feeding method has the grounding conductor connected to the beginning and end of each potential group.

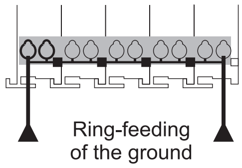


Fig. 2-25: Ring-feeding g0xxx07e

Attention
The regulations relating to the place of assembly as well as the national regulations for maintenance and inspection of the grounding protection must be observed.

2.9 Shielding (Screening)

2.9.1 General

The shielding of the data and signal conductors reduces electromagnetic interferences thereby increasing the signal quality. Measurement errors, data transmission errors and even disturbances caused by overvoltage can be avoided.

Attention
Constant shielding is absolutely required in order to ensure the technical specifications in terms of the measurement accuracy.

The data and signal conductors should be separated from all high-voltage cables.

The cable shield should be potential. With this, incoming disturbances can be easily diverted.

The shielding should be placed over the entrance of the cabinet or housing in order to already repel disturbances at the entrance.

2.9.2 Bus Conductors

The shielding of the bus conductor is described in the relevant assembly guidelines and standards of the bus system.

2.9.3 Signal Conductors

Bus modules for most analog signals along with many of the interface bus modules include a connection for the shield.

Note
For a better shield performance, the shield should have previously been placed over a large area. The WAGO shield connection system is suggested for such an application.
This suggestion is especially applicable if the equipment can have even current or high impulse formed currents running through (for example initiated by atmospheric discharge).

2.9.4 WAGO Shield (Screen) Connecting System

The WAGO Shield Connecting system includes a shield clamping saddle, a collection of rails and a variety of mounting feet. Together these allow many different possibilities. See catalog W4 volume 3 chapter 10.

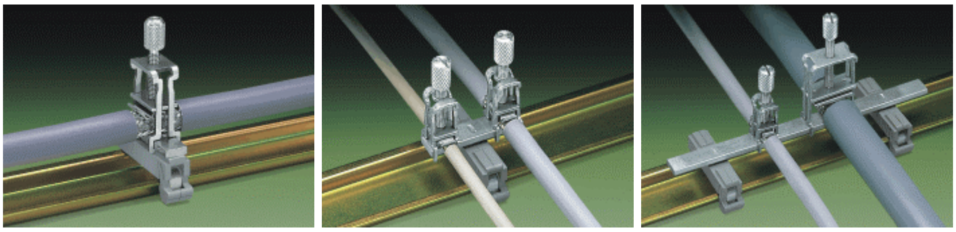


Fig. 2-26: WAGO Shield (Screen) Connecting System p0xxx08x, p0xxx09x, and p0xxx10x

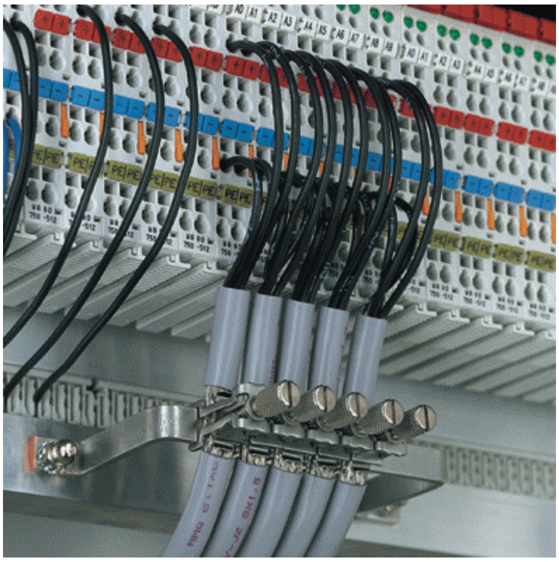


Fig. 2-27: Application of the WAGO Shield (Screen) Connecting System p0xxx11x

2.10 Assembly Guidelines/Standards

DIN 60204,	Electrical equipping of machines
DIN EN 50178	Equipping of high-voltage systems with electronic components (replacement for VDE 0160)

3 Fieldbus Controller

3.1 BACnet/IP Controller 750-830

3.1.1 Description

The 750-830 BACnet Controller connects the WAGO-I/O-SYSTEM with the BACnet protocol. The 750-830 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.

3. **Application client:** Using the client functionality, objects and their properties can be accessed by other BACnet devices.

Access to BACnet/IP networks is provided by the controller's RJ45 interface. The RS232 interface can be used as a standard RS232 or (beginning with software version 2) as a BACnet-PTP connection to other PTP-capable BACnet devices.

The 750-830 BACnet/IP Controller is based on an ETHERNET controller and supports the corresponding functions:

All input signals from the sensors are combined. After connecting the controller, all of the I/O modules on the node are determined and a local process image is created from these. Analog and specialty module data is sent via words and/or bytes; digital data is sent bit by bit.

The local process image is divided into two data zones containing input and output data areas. The data of the analog modules are written into the process image in the order of their position after the controller.
The bits of the digital modules are combined into words and then also mapped onto the analog modules in the process image. If the number of digital inputs and outputs is greater than 16 bits, the controller automatically begins a new word.

According to IEC -61131-3 programming, the processing of the process data occurs on location in the PFC. The link results created by this can be output directly to the actors or transmitted via the bus to the higher-order controller.

The controller can then optionally communicate with higher-order systems either via 10/100 Mbit/s (ETHERNET), "100BaseTX" or "10BaseT".

An application program can be created using the WAGO-I/O-PRO CAA software, based on IEC 61131-3. The controller provides 512 KB of program memory, 256 KB of data memory and 24 KB of retain memory for this purpose. Start-up and configuration of the BACnet/IP Controller is performed using the Windows-compliant WAGO BACnet Configurator.

For communication via BACnet, the BACnet/IP and BACnet/PTP protocols are supported. Process data are also sent via MODBUS/TCP(UDP). HTTP, BootP, DHCP, DNS, SNTP, FTP, SNMP V1 and SMTP are used for administration and diagnosis.

The programmer 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 (resolution 1 second), 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, i.e. several programs can be executed semi-simultaneously.

An internal server is available for Web-based applications.
Information on configuration is also stored as HTML pages in the fieldbus controller and can be read using a customary web browser (web-based Management-System). In addition, internal HTML pages can also be stored using an implemented file system.

3.1.2 Compatibility

Additional Information
To get the current software version for programming and configuring the Controller 750-830, go to our website at http://www.wago.com → Service → Documentation → WAGO-I/O-SYSTEM759 → WAGO-I/O-PRO

Additional Information
You can find the BACnet Configurator on the internet on the website http://www.wago.com Service → Downloads → Building Automation → BACnet Downloads.
The documentation for the BACnet Configurator can be found under Service → Documentation → WAGO-I/O-SYSTEM 750 → Fieldbus Coupler and Programmable Fieldbus Controller → 750-830.

3.1.3 Hardware

3.1.3.1 View

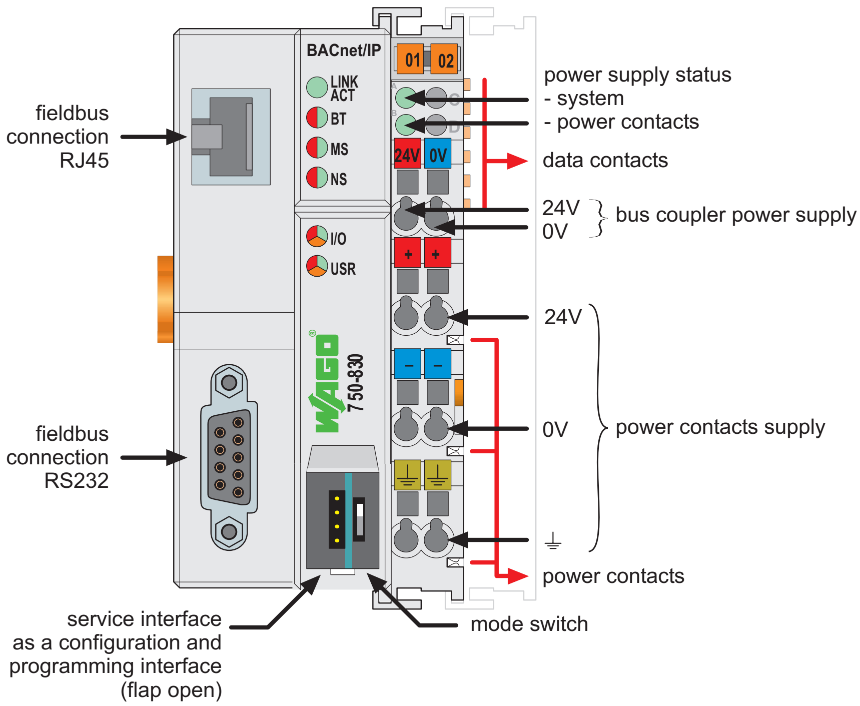


Fig. 3-1: BACnet/IP Controller G083000e

The controller in detail:

- A power supply unit for the system supply and power jumper contacts for the field supply via bus modules.
- Two fieldbus connections (RJ45 and RS232)
- LEDs as status display of the operation, the bus communication, the operating voltages as well as for error messages and diagnostics
- Service interface, alternative for programming and configuration (see 3.1.3.5)
- Operating mode switch (see 3.1.3.6)

3.1.3.2 Power Supply

The power supply is derived from modules with CAGE CLAMP® connections. 24 V power supply (see Fig. 3-1) for system power and power to the field side.

The integrated power supply provides the required power to the electronics and the bus modules.

An electrically isolated power supply is provided to the fieldbus interface.

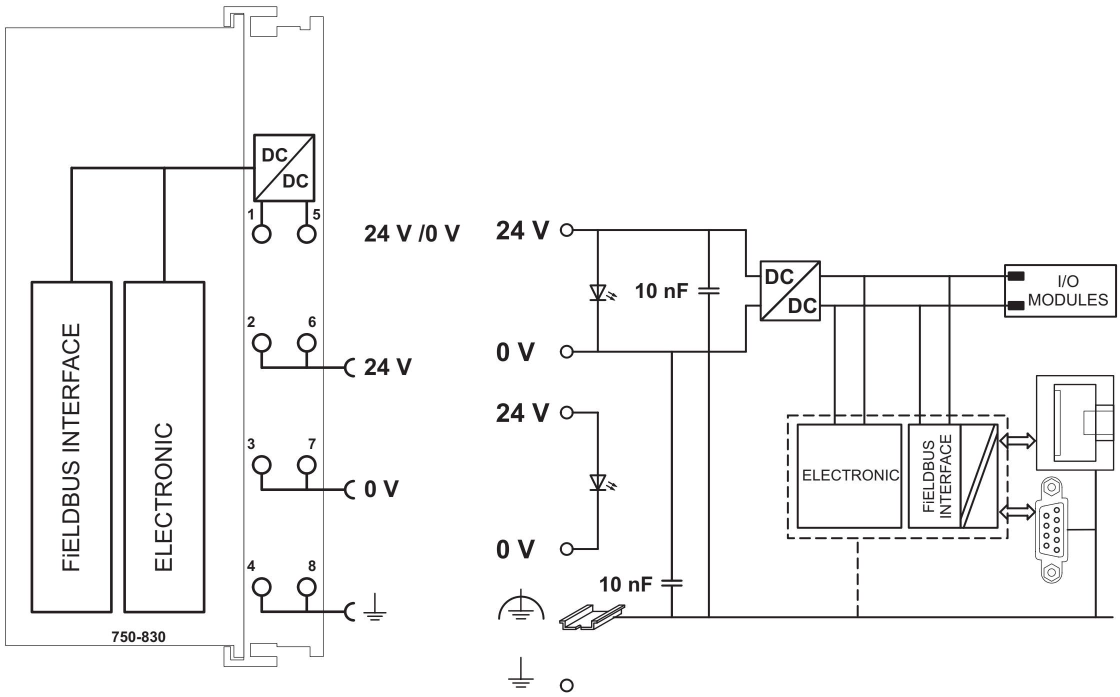
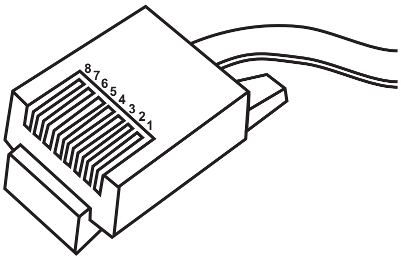


Fig. 3-1: Power Supply G083001e

3.1.3.3 Fieldbus Connection

The connection to the fieldbus is made via an RJ45 connector, which is also called a "Western plug." Wiring for the RJ45 socket on the fieldbus controller adheres to 100BaseTX specifications. It is mandatory to use a twisted pair cable of category 5 as a connecting cable. Cable types S-UTP (Screened Unshielded Twisted Pair) and STP (Shielded Twisted Pair) with a maximum segment length of 100 m can be used.



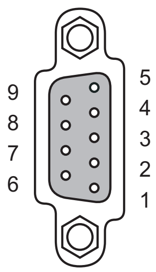
Tab. 3-1: Plug-in Contacts

Contact	Signal	
1	TD +	Transmit Data +
2	TD -	Transmit Data -
3	RD +	Receive Data +
4		not assigned
5		not assigned
6	RD -	Receive Data -
7		not assigned
8		not assigned

The connection point is lowered for mounting into an 80 mm-high switchgear cabinet after the connector attachment.

Attention:
The bus connection is approved **only** for use in LAN networks, not for the connection of telecommunication lines!

In addition to BACnet/IP, the controller also supports the BACnet/PTP protocol. The controller can communicate, when connected to the RS232 interface, with other PTP-capable devices through a modem connection (half router). The SUB-D connector for the RS232 interface is wired as follows:



View of the Front of the Socket

Tab. 3-2: Bus Connection and Connection Plug Arrangement for the RS232

Contact	Signal		Description
1	-	-	Not used
2	RxD	Receive Data	Receive signal
3	TxD	Transmit Data	Transmit signal
4	-	-	Not used
5	GND	Ground	Signal and supply ground
6	-	-	Not used
7	RTS	Request to send	Request to send; logical Zero = ready for data receipt
8	CTS	Clear to send	Send readiness; logical Zero = ready for sending data
9	-	-	Not used

The pin arrangement corresponds to the RS232 DCE arrangement. This allows the use of customary 9-pole 1:1 socket/plug cables for the direct connection of a PC.

3.1.3.4 Indicators

The operational status for the fieldbus controller and the node is indicated by light emitting diodes (LEDs).
These are multi-colored (red, green or red-green (=orange)).

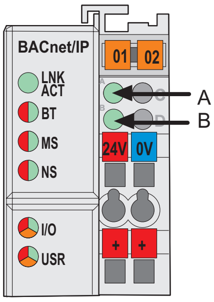


Fig. 3-2: Indicators 750-830 g083002x

Tab. 3-3: LED Signals

LED	Color	Meaning
LNK/ACT	Off Green Green flashing	Network connection and activity at Port 1
BT	Off Green Green flashing Red	BACnet data traffic
MS	Off Green Green flashing Red, flashing Red Red-green (orange) flashing	System operating mode
NS	Off Green Green flashing Red flashing Red Red-green (orange) flashing	IP address configuration and communication
I/O	Red/green (orange)	The "I/O" LED indicates the operational status of the node and signals any errors.
USR	Red/green (orange)	The "USR" LED can be controlled by a user program.
A	Green	Status of the system power supply
B	Green	Status of the power supply for the power contacts

Additional Information
The evaluation of the LED signals is described in more detail in section 3.1.9, "LED Signals".

3.1.3.5 Configuration Interface and Programming Interface

The configuration interface is located behind the cover flap. It is used for communication with WAGO-I/O-CHECK, WAGO-I/O-PRO CAA and for downloading firmware.

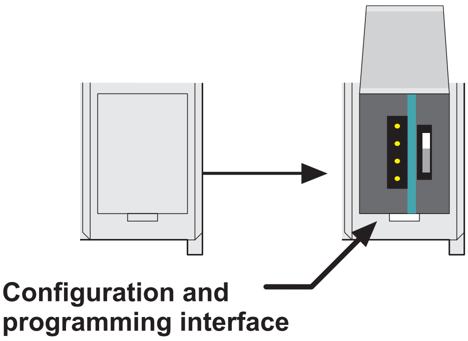


Fig. 3-3: Configuration Interface g01xx07e

The communication cable (750-920) is connected to the four-pole header.

Notice
The 750-920 Communication cable may not be connected or removed when the system is energized; i.e., there must be no power to the coupler/controller!

3.1.3.6 Mode Selector Switch

The mode selector switch is located behind the cover flap.

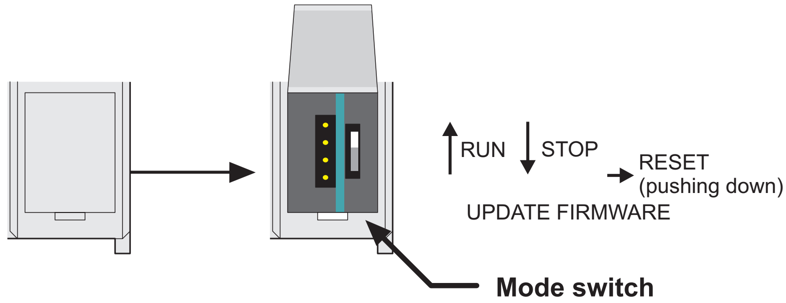


Fig. 3-4: Mode Selector Switch g01xx10e

The switch is a push button or sliding switch with three positions and a push-button function.

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

Tab. 3-4: Functions of the Mode Selector Switch

Mode Selector Switch	Function
From center to top position	Firmware and PFC applications will be executed (activate program processing/“RUN“)
From top to center position	Firmware will be executed; PFC application halted (stop program processing/“STOP“)
Lower position	Controller starts the operating system loader
Pressed down (e.g., using a screwdriver)	Hardware reset All outputs and flags are reset; variables are set to 0 or to FALSE or to an initial value. Retain variable or flags are not changed. A hardware reset can be performed either at STOP or at RUN at any position of the mode selector switch!

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

Note
The position of the mode selector switch is not important when starting or stopping the PFC application from WAGO-I/O-*PRO* CAA.

Attention
Remember that if outputs are set when switching the mode selector switch from "RUN" to "STOP" that these will remain set! Software-side switch offs, e.g. by initiators, are ineffective, because the program is no longer processed.

Note
The user has the opportunity to define the status of the outputs for STOP. A web page is then opened via the "PLC" link in the web-based management system where the function can be defined accordingly (see section 3.1.8.7). If there is a check mark in the box "Enabled", all outputs will be set to zero; if there is no check mark, the outputs will retain their last current value.

3.1.3.7 Hardware Address (MAC ID)

Each WAGO BACnet/IP Controller has a unique and internationally unambiguous physical address, referred to as the MAC-ID (Media Access Control Identity). This is located on the rear of the controller and on a self-adhesive tear-off label on the side of the controller. The MAC ID has a set length of 6 bytes (48 bits) (hexadecimal). The first three bytes identify the manufacturer (e.g. 00:30 DE for WAGO). The second 3 bytes indicate the consecutive serial number for the hardware.

3.1.4 Operating System

3.1.4.1 Boot-up

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

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

During the initialization phase, the fieldbus controller detects the bus 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 controller switches to the status "RUN". The "I/O" LED then stays lit continuously in green.

3.1.4.2 PFC Cycle

After successful boot-up, the PFC cycle starts if the mode selection switch is in the upper position or due to a start command from the WAGO-I/O-PRO CAA. The input and output data for the field bus, bus modules and the timer values are read. Subsequently the PFC program in the RAM is processed; after that, the output data of the fieldbus and the bus modules are written in the process image. Operating system functions, among others, for diagnostics and communication are performed and the values of the timer are updated at the end of the PFC cycle. The cycle starts again with the 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 therefore the PFC cycle time as well, 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.

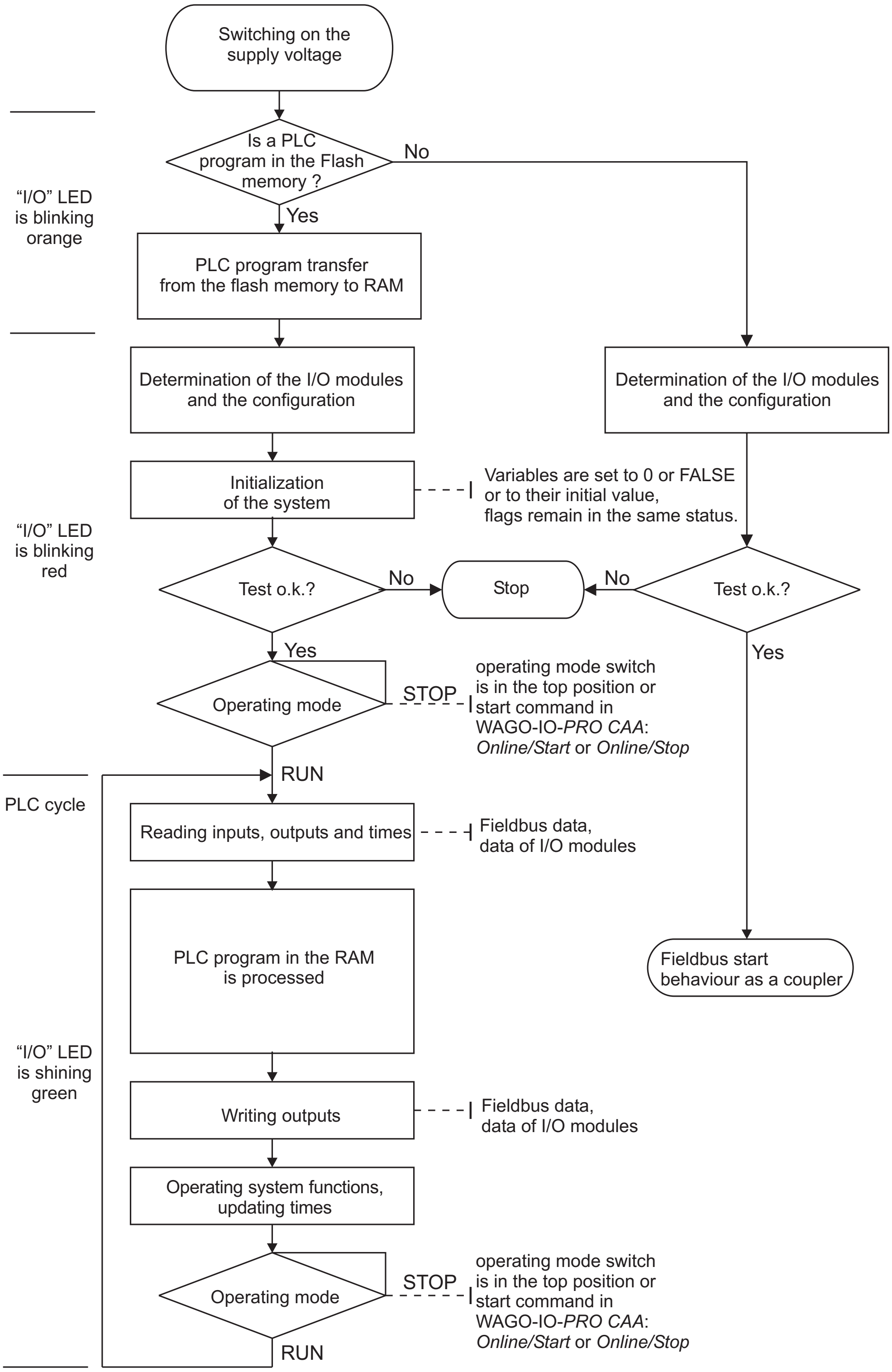


Fig. 3-5: Controller operating system g015041e

3.1.5 Process Image

Sections 3.1.5 and 3.1.6 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.

3.1.5.1 Basic Structure

After switching on, the controller identifies all bus modules connected with the node that send or receive data (data width/bit width > 0).
A node can consist of a mixed arrangement of analog and digital modules.

Note
Using the WAGO 750-628 Internal Data Bus Extension Coupler Module and 750-627 End Module makes it possible to connect up to 250 modules to the 830-830 BACnet/IP Controller.

Attention
You can find the number of the input and output bits or bytes of the individually switched on bus modules in the corresponding descriptions of the bus modules.

The controller creates an internal local process image on the basis of the data width, the type of bus module and the position of the module in the node. This is divided into an input and an output area.

The data of the digital bus modules is bit-oriented; i.e., digital data is sent bit by bit. Analog bus modules represent all byte-oriented bus modules, which send data byte by byte. Counter modules, DALI, MP bus, EnOcean and communication modules, for example, are included in this group of bus modules.

For both the local input and the output process image, the bus module data is stored in the corresponding process image according to the order in which the modules are connected to the controller.
First, all the byte-oriented (analog) bus modules are filed in the process image, then the bit-oriented (digital) bus modules. The bits of the digital modules are assembled into bytes. If the number of digital inputs and outputs is greater than 8 bits, the controller automatically begins a new byte.

Note
If a node is changed or expanded, this may result in a new process image structure. The process data addresses would then change. In case of an expansion, the process data of all previous modules has to be taken into account.

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

For the image of the MODBUS/PFC variables, the memory range of words 256 ... 511 is reserved, meaning the image for the MODBUS/PFC variables is created behind the process image for the bus 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.

For all WAGO fieldbus controllers, access by the PLC (CPU) to process data is made regardless of the fieldbus system; 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 the MODBUS functions that are implemented.

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

3.1.5.2 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.

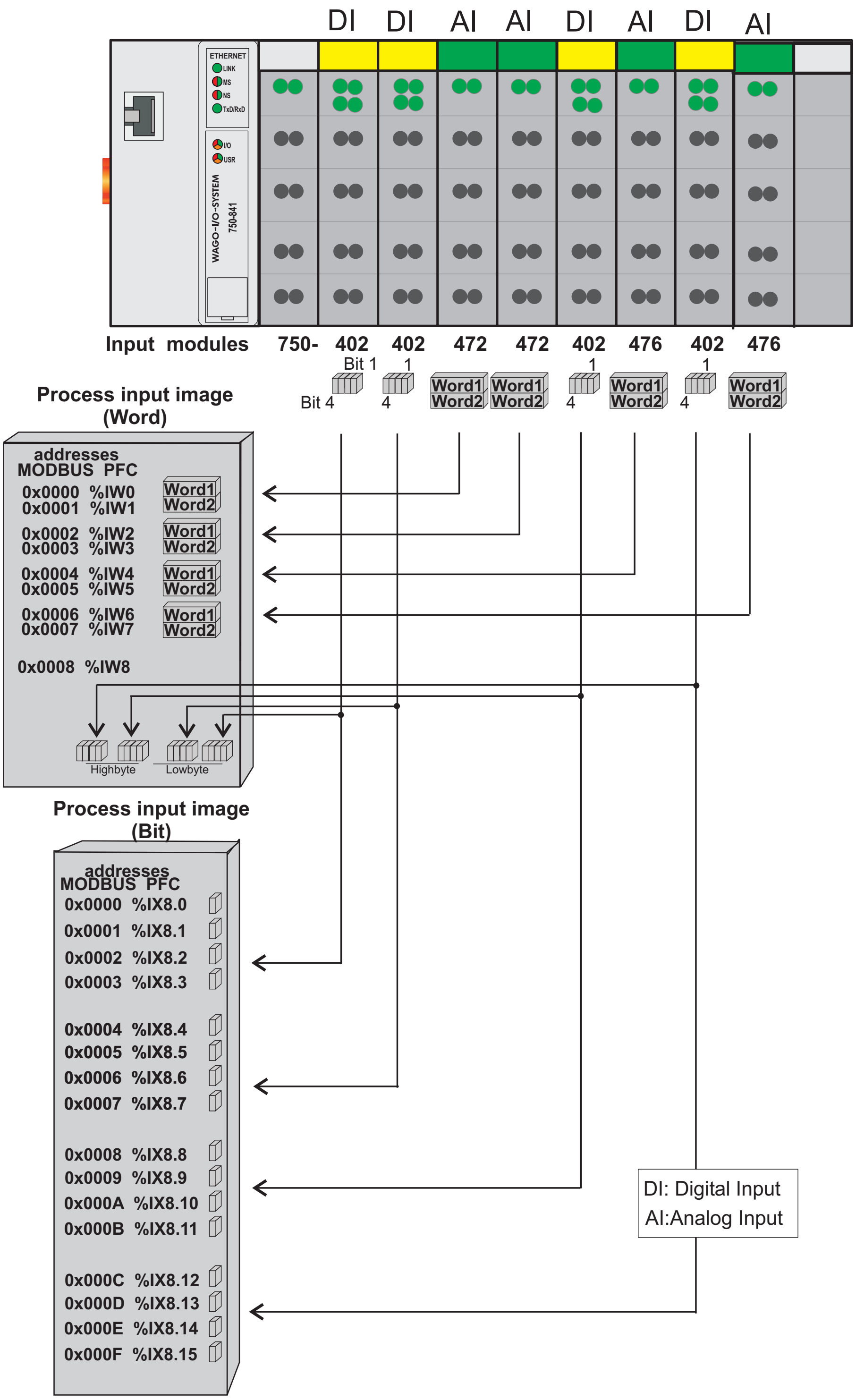


Fig. 3-6: Example of process image for input data G015024e

3.1.5.3 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
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.

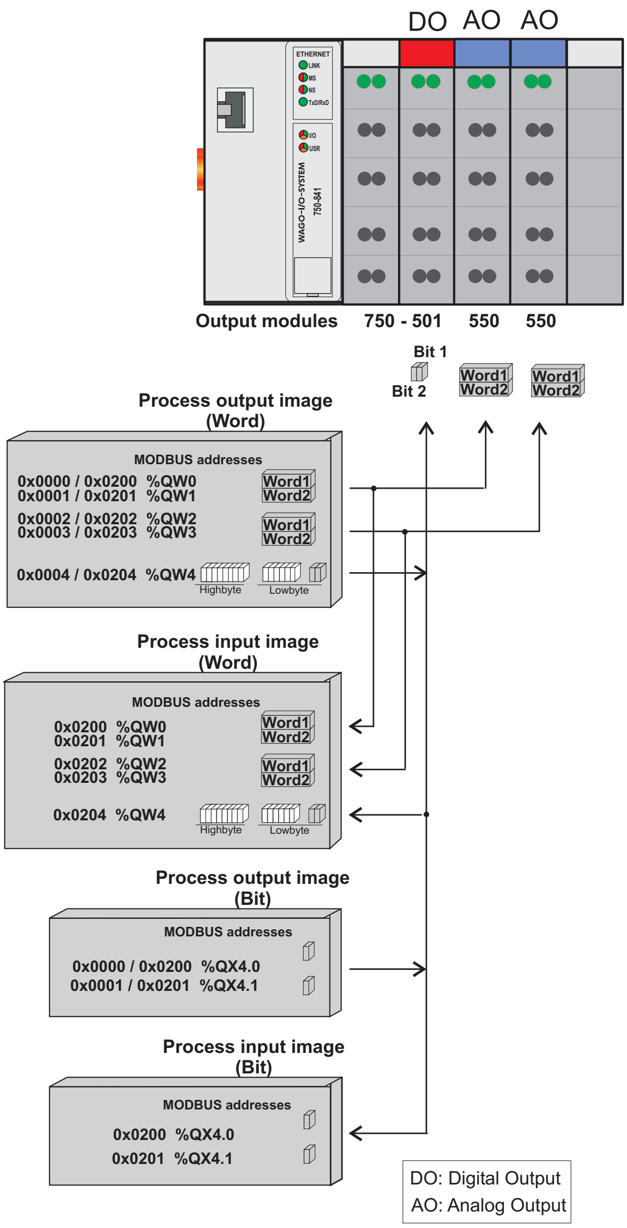


Fig. 3-7: Example of process image for output data G015025e

3.1.5.4 MODBUS Process Data

For some bus modules and their different versions, 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 Intel formats. The modules can be mapped directly via addresses with MODBUS.

Additional Information
For the fieldbus-specific structure of the process values of all bus modules within the 750 and 753 Series of the WAGO-I/O-SYSTEM, refer to section 3.1.5.4, "MODBUS Process Data".

3.1.6 Data Exchange

Exchange of process data takes place with BACnet/IP controllers 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 establish a defined number of simultaneous connections (socket connections) to other network subscribers:

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

The maximum number of simultaneous connections cannot 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)
- the interface to the bus modules

There is a data exchange between the fieldbus master and the bus modules, between the PLC function of the PFC (CPU) and the bus modules, and between the fieldbus master and PLC function of the PFC (CPU). If the MODBUS master is used as the fieldbus, access is made to the data using a MODBUS function implemented in the controller.
Data access by the PFC is carried out with the aid of an IEC 61131-3 application program. Data addressing varies greatly here.

3.1.6.1 Memory Areas

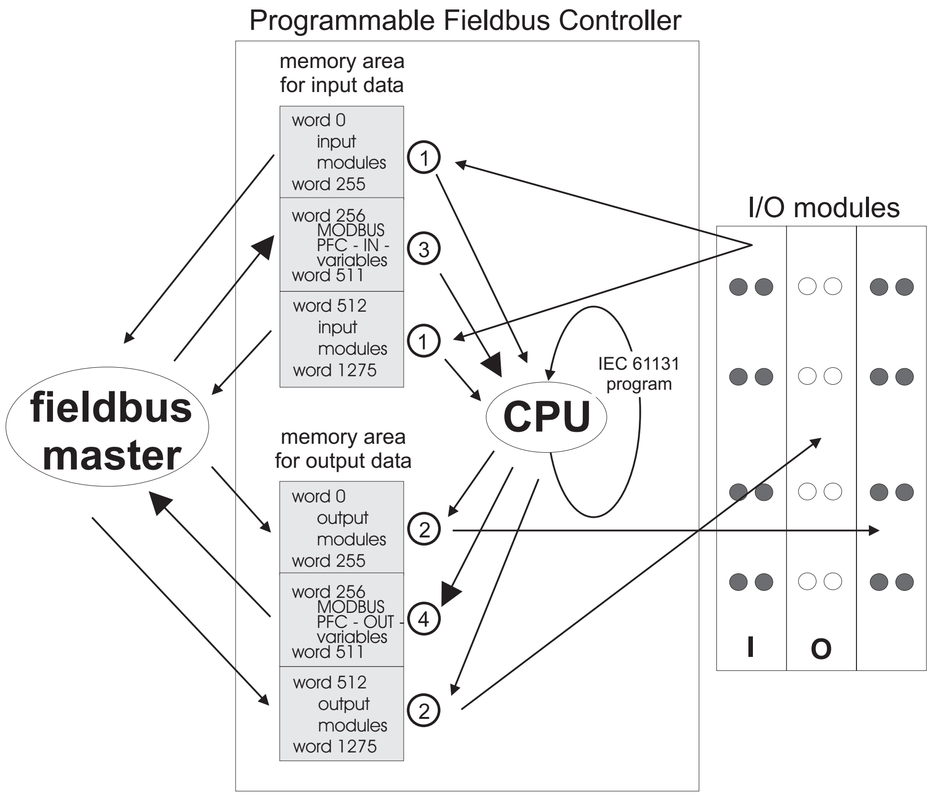


Fig. 3-8: Memory areas and data exchange g015038d

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

① The data of the input modules can be read by the CPU and from the fieldbus side.

② In the same manner, writing to the output modules is possible from the CPU and from 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 via the IEC -61131-3 program are filed in the output memory space and can be read out by the master.

The memory area for word 1276 ... 1531 is adjacent to the physical bus module data. This area is reserved and may not be used by the user. The subsequent memory area, starting from word 1532, is reserved for future protocol expansion and other PFC variables.

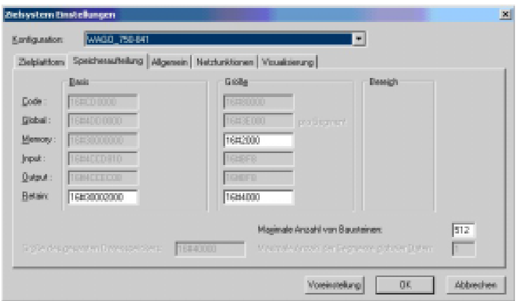
In addition, all output data is mirrored in the BACnet/IP controller to a memory area with the address offset 0x0200 and 0x1000. This makes it possible to read back output values by adding 0x0200 and 0x1000 to the MODBUS address. Other memory areas are also provided in the controller, some of which cannot be accessed by the fieldbus side, however.

- **Data memory (256 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 (512 kByte)**
  The IEC 61131-3 program is filed in the program memory. The code memory is a flash memory. Once the supply voltage is applied, the program is transmitted 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 CAA.

- **NOVRAM remanent memory (24 kByte)**
  The remanent memory is a non-transitory memory, i.e. all values are retained following a power failure. The memory management is automatic. The 24 kByte memory area is normally divided into an 8 kByte addressable range for flags (%MW0 ... %MW 4095) and a 16 kByte retain area for variables without memory area addressing, or for variables that are explicitly defined by "var retain".

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



The start address for the flag area is fixed at 16#30000000. 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#2000, followed by the retain memory, with the start address 16#30002000 and the size 16#4000.

3.1.6.2 Addressing

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

Note
This section explains addressing and internal functioning of a controller with connected modules in more detail. It is essential that you understand these correlations in order to conduct conventional addressing by counting (MODBUS).
If you use MODBUS, take care that "fieldbus 1" is chosen in the WAGO-I/O-PRO CAA (see section 3.1.8.1).

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 only the connected modules in the I/O Configurator. The software then takes care of correct addressing (see Fig. 3-1).).

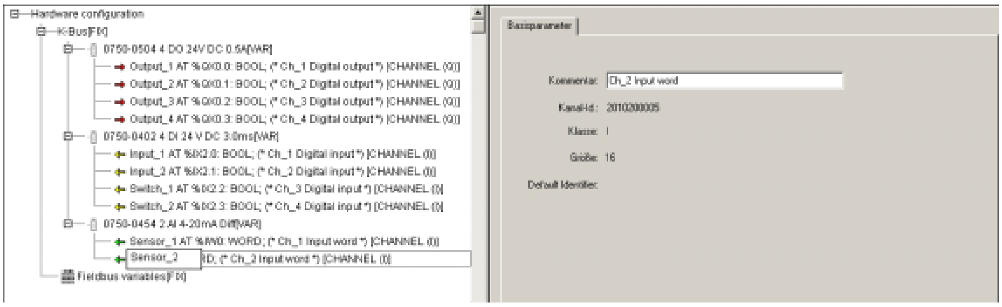


Fig. 3-1: WAGO-I/O Configurator

The I/O Configurator is started from the WAGO-I/O-*PRO* CAA. For more details, refer to Section 3.1.8.1.

3.1.6.2.1 Addressing of Bus Modules

Addressing first references complex modules (modules that occupy several bytes) in accordance with their physical order downstream of the fieldbus 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 as a complete byte is occupied by the bit-oriented modules, the process begins automatically with the next byte.

Note
For the meaning of input and output bits or bytes of the individual bus module please refer to the corresponding bus module description.

Note
If a node is changed or expanded, this may result in a new process image structure. In this case, the process data addresses also change. In case of an expansion, the process data of all previous modules has to be taken into account.

Tab. 3-5: Data width for bus modules

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

3.1.6.2.2 Example of Addressing

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 to one controller. The final element is an end module that is not taken into account for addressing.

Tab. 3-6: Example of addressing

Count Sequence	Module	Function	Data Width	Hardware Address
1.	750-467	2 AI / 0-10 Volt	2 x 16 Bit	%IW0 and %IW1
2.	750-400	2 DI	2 x 1 Bit	%IX2.0 and %IX2.1
3.	750-550	2 AO / 0-10 Volt	2 x 16 Bit	%QW0 and %QW1
4.	750-501	2 DO	2 x 1 Bit	%QX2.0 and %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. Hardware addresses are then **%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 a**nd **%QX2.1**.

Note
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.

3.1.6.2.3 Address Ranges

Subdivision of the address ranges for word-by-word addressing in accordance with IEC61131-3:

Tab. 3-7: Breakdown of address range

Word	Data
0-255	Physical bus modules
256-511	MODBUS-PFC variables
512-1275	Other physical bus modules

Word 0-255: First address range for the input/output data of the bus module:

Tab. 3-8: Address range, word 0 - 255

Data Width	Address								
Bit	0.0 ...	0.8... 0.15	1.0 ... 1.7	1.8... 1.15	.....	254.0 ... 254.7	254.8... 254.15	255.0 ... 255.7	255.8... 255.15
Byte	0	1	2	3	.....	508	509	510	511
Word	**0**		**1**		**.....**	**254**		**255**	
DWord	0				.....	127			

Word 256-511: Address range for the MODBUS/TCP fieldbus data:

Tab. 3-9: Address range, word 256 - 511

Data Width	Address								
Bit	256.0 ... 256.7	256.8 ... 256.15	257.0 ... 257.7	257.8 ... 257.15	.....	510.0 ... 510.7	510.8 ... 510.15	511.0 ... 511.7	511.8 ... 511.15
Byte	512	513	514	515	.....	1020	1021	1022	1023
Word	**256**		**257**		**.....**	**510**		**511**	
DWord	128				.....	255			

Word 512-1275: Second address range for the input/output data of the bus module:

Tab. 3-10: Address range, word 512 - 1275

Data Width	Address								
Bit	512.0. 512.7	512.8... 512.15	513.0 .. 513.7	513.8... 513.15	.....	1274.0.. 1274.7	1274.8.. 1274.15	1275.0 ... 1275.7	1275.8... 1275.15
Byte	1024	1025	1026	1027	.....	2548	2549	2550	2551
Word	**512**		**513**		**.....**	**1274**		**1275**	
DWord	256				.....	637			

Address range for flags:

Tab. 3-11: Address range for flags

Data Width	Address								
Bit	0.0 ... 0.7	0.8... 0.15	1.0... 1.7	1.8... 1.15	.....	12287.0.. 12287.7	12287.8.. 12287.15	12288.0 ... 12288.7	12288.8... 12288.15
Byte	0	1	2	3	.....	24572	24573	24574	24575
Word	**0**		**1**		**.....**	**12287**		**12288**	
DWord	0				.....	6144			

IEC 61131-3 Overview of Address Areas:

Tab. 3-12: IEC 61131-3 address areas

Address Area	MODBUS access	PLC Access	Description
phys. inputs	read	read	Physical inputs (%IW0 ... %IW255 and %IW512 ... %IW1275)
phys. outputs	read/write	read/write	Physical outputs (%QW0 ... %QW255 and %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 tab	read/write	---	see "ETHERNET" section
Firmware register	read	---	see "ETHERNET" section
Retain variables	read/write	read/write	Remanent memory (%MW0 ... %MW12288)

3.1.6.2.4 Absolute Addressing

Direct presentation of individual memory cells (absolute addresses) based on IEC 1131-3 is performed using character strings:

Tab. 3-13: Absolute addresses

Position	Prefix	Designation	Commentary
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) Double word (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
The character strings for absolute addresses must be entered connected, i.e. without spaces or special characters!

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)	

	Flag			
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 2nd byte: 2 x word address + 1
DWord address	Word address (even number) / 2 or Word address (uneven number) / 2, rounded

3.1.6.3 Data Exchange between MODBUS/TCP Master and Bus Modules

Data exchange between the MODBUS/TCP master and the bus 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 four (4) different types of process data in the controller:

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

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

Tab. 3-14: 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
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.

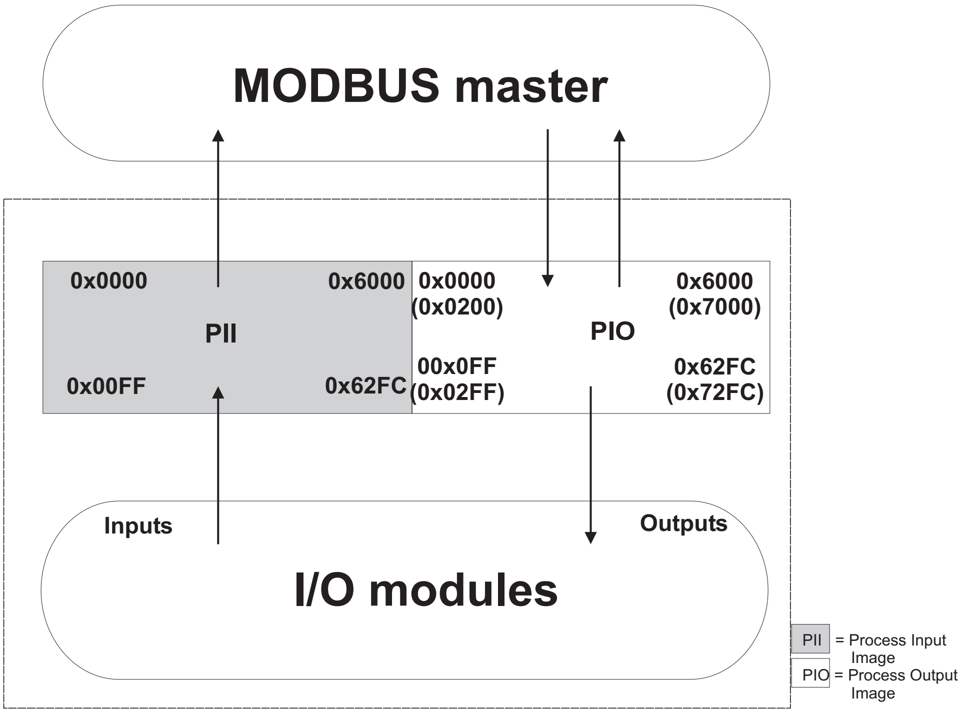


Fig. 3-9: Data exchange between MODBUS Master and bus modules g015045e

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.

3.1.6.4 Data Exchange between PLC Function (CPU) and Bus Modules

The PLC function (CPU) of the PFC uses absolute addresses to access the bus module data directly.

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 remanent memory area in this process. The results of linking can then be written directly to the output data employing absolute addressing.

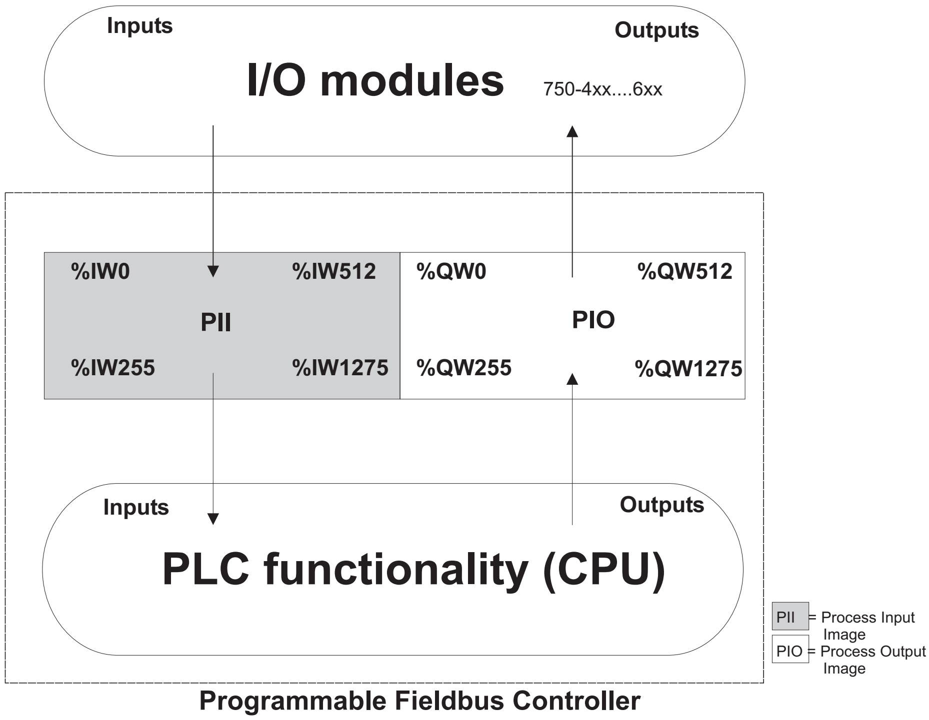


Fig. 3-10: Data exchange between PLC function (CPU) of the PFC and the bus modules
g015043d

3.1.6.5 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 is routed as input variables to the PFC, where it is further processed.
Data created in the PFC is transmitted via the fieldbus to the master as output variables.

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

3.1.6.5.1 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 bus 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 by using the following formula:

BitNo = (word * 16) + BitNo. 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 <WordNo.>.<BitNo.> = 256.1

The PLC function of the PFC can also access data by bytes and by double-word access.

Addresses are calculated based on the following equations for byte-based access:
High-byte address = Word address*2
Low-byte address = (Word address*2) + 1

Addresses are calculated according to the following equation for double-word-based access:
Double-word address = High word address/2 (rounded down)
or = Low word address/2

3.1.6.5.2 Juxtaposition of MODBUS/TCP and IEC 61131-3 Addresses

3.1.6.5.2.1 Word Access

Tab. 3-15: Word access

Method	MODBUS Addresses decimal	hexadecimal	IEC 61131 Addresses	Description
FC3 - Read Multiple Register FC4 – Read Holding Register	0... 255	0x0000 – 0x00FF	%IW0... %IW255	phys. inputs (1)
	256... 511	0x0100 – 0x01FF	%QW256... %QW511	PFC-OUT variables
	512 ... 767	0x0200 – 0x02FF	%QW0... %QW255	phys. outputs (1)
	768 ... 1023	0x0300 – 0x03FF	%IW256... %IW511	PFC-IN variables
	illegal address	0x0400 – 0x0FFF	not supported	
	4096... 8191	0x1000 – 0x1FFF	not supported	Configuration tab
	8192 ... 12287	0x2000 - 0x2FFF	not supported	Firmware register
	12288... 13385	0x3000 - 0x3FFF	%MW0... %MW4095	Flag area (Default: 8 kByte, size variable)
	24576 ... 25340	0x6000- 0x62FB	%IW512... %IW1275	phys. inputs (2)
	28672 ... 29436	0x7000- 0x72FB	%QW512... %QW1275	phys. outputs (2)
FC16 – Write Multiple Register	0... 255	0x0000 – 0x00FF	%QW0... %QW255	phys. outputs (1)
	256... 511	0x0100 – 0x01FF	%IW256... %IW511	PFC-IN variables
	512... 767	0x0200 – 0x02FF	%QW0... %QW255	phys. outputs (1)
	768 ... 1023	0x0300 – 0x03FF	%IW256... %IW511	PFC-IN variables
	illegal address	0x0400 – 0x0FFF	not supported	
	4096... 8191	0x1000 – 0x1FFF	not supported	Configuration tab
	illegal address	0x2000 - 0x2FFF	not supported	Firmware register
	12288... 16383	0x3000 - 0x3FFF	%MW0... %MW4095	Flag area (Default: 8 kByte, size variable)
	24576 ... 25339	0x6000- 0x62FB	%QW512... %QW1275	phys. outputs (2)
	28672 ... 29435	0x7000- 0x72F	%QW512... %QW1275	phys. outputs (2)

3.1.6.5.2.2 Bit Access

Tab. 3-16: Bit access

Method	MODBUS Addresses decimal	MODBUS Addresses hexadecimal	IEC 61131-Addresses	Description
FC2 - Read Input Discrete FC1 = FC2 + 0x0200 – Read Coils	0... 511	0x0000 – 0x01FF	%IX(DigitalOffSet + 0).0 ... %IX(DigitalOffSet + 31).15	phys. inputs (1)
	512... 1023	0x0200 – 0x03FF	%QX(DigitalOffSet + 0).0 ... %QX(DigitalOffSet + 31).15	phys. outputs (1)
	Illegal address	0x0400 – 0x0FFF	not supported	
	4096 … 8191	0x1000 – 0x1FFF	%QX256.0 ... %QX511.15	PFC-OUT variables
	8192 ... 12287	0x2000 – 0x2FFF	%IX256.0 ... %IX511.15	PFC-IN variables
	12288 ... 32767	0x3000 - 0x7FFF	%MX0.0 ... %MX1279.15	Flag area (Default: 8 kByte, size variable)
	32768 ... 34295	0x8000 - 0x85F7	%IX32.0 … %IX127.7	phys. inputs (2)
	36864 ... 38391	0x9000 - 0x95F7	%QX32.0 .. %QX127.7	phys. outputs (2)
FC15- - Force Multiple Coils	0... 511	0x0000 – 0x01FF	%QX(DigitalOffSet + 0).0 ... %QX(DigitalOffSet + 31).15	phys. outputs (1)
	512... 1023	0x0200 – 0x03FF	%QX(DigitalOffSet + 0).0 ... %QX(DigitalOffSet + 31).15	
	Illegal address	0x0400 – 0x0FFF	not supported	
	4096... 8191	0x1000 – 0x1FFF	%IX256.0 ... %IX511.15	PFC-IN variables
	8192... 12287	0x2000 – 0x2FFF	%IX256.0 ... %IX511.15	
	12288... 32767	0x3000 - 0x7FFF	%MX0.0... %MX1279.15	Flag area (Default: 8 kByte, size variable)
	32768... 34295	0x8000 - 0x85F7	%QX32.0 .. %QX1275.15	phys. outputs (2)
	36864 ... 38391	0x9000 - 0x95F7	%QX32.0 .. % QX127.7	

3.1.6.5.2.3 Application Example

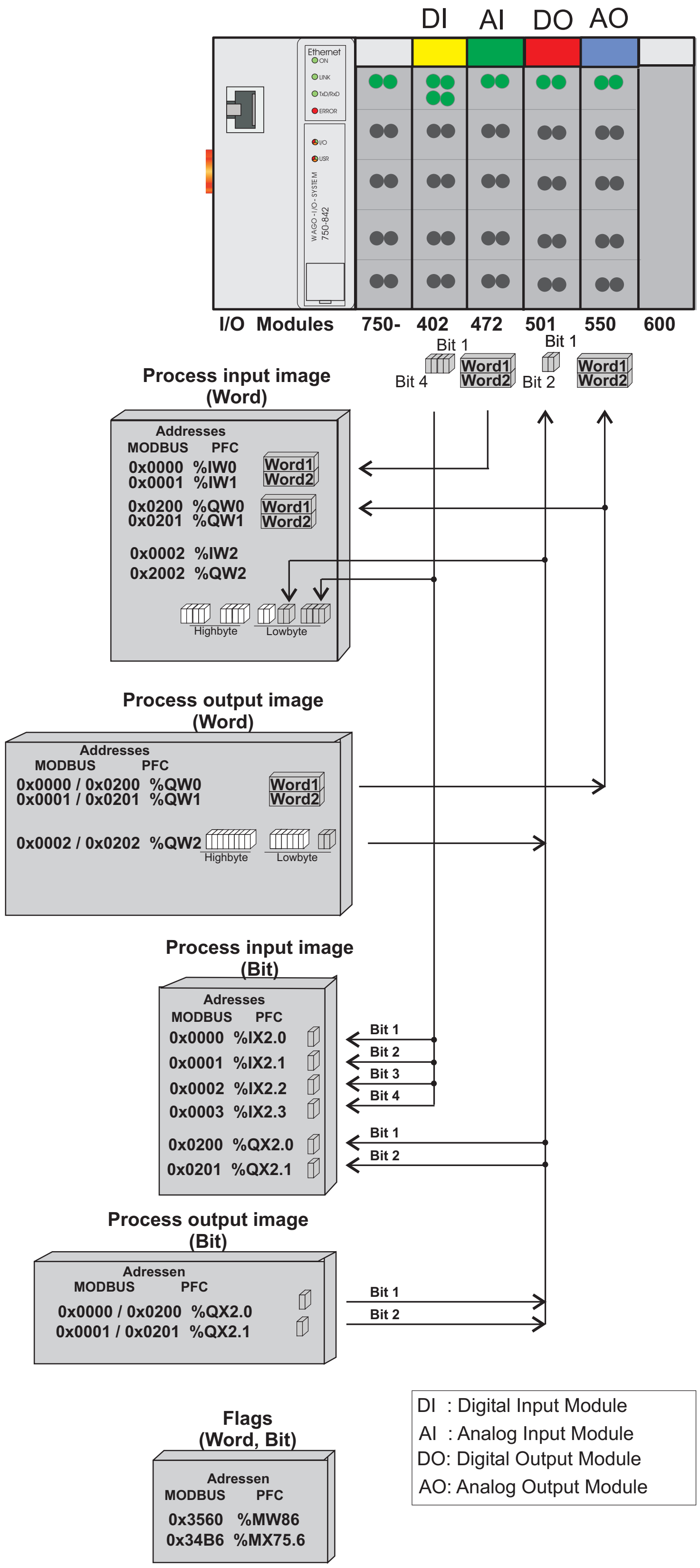


Fig. 3-11: Addressing example for a fieldbus node

g012948e

3.1.7 Fieldbus Node Start-up

This chapter provides a step-by-step description of how to start-up a BACnet fieldbus node. The controller must be assigned an IP address before it can communicate properly, which can be done in one of two way:

- **3.1.7.1:** Startup using **WAGO-ETHERNET-Settings**
  Assigning of IP addresses via the serial communication port

- **3.1.7.2:** Startup using the WAGO-BootP server
  Assigning of IP addresses via fieldbus, but with more steps being required here than for method 3.1.7.1.

Additional Information
By default, the IP address is assigned to the BACnet/IP controller through a BootP server.

3.1.7.1 Startup Using the WAGO ETHERNET Settings

"WAGO-ETHERNET Settings" (759-316) is a Windows application software that can be used to read and edit bus-specific parameters of WAGO-ETHERNET couplers/controllers. Communication cables or WAGO radio-link adapters can be used for data communication.

The following steps are included:

- Connecting of client PC and fieldbus nodes (see section 3.1.7.1.1)
- Assigning of IP addresses to the fieldbus nodes (see Section 3.1.7.1.2)
- Testing for proper functioning of the fieldbus nodes (see Section 3.1.7.1.3)

3.1.7.1.1 Connecting Client PC and Fieldbus Nodes

1. Connect the installed BACnet fieldbus node with the configuration and programming port of the controller using the communication cable 750-920 to a vacant serial port on your computer.

The client PC does not require a network card when connected directly, at a serial port.

Note
The communication cable 750-920 may not be connected or removed when the system is energized; i.e., there must be no power to the coupler/controller!

After the power is switched on, the controller is initialized. The fieldbus controller determines the bus module configuration and creates a process image. During startup, the "I/O" LED will flash red rapidly.

After a brief period, the 'I/O' LED lights up green, meaning the fieldbus controller is operational. If an error occurred during start-up, an error code is indicated by a red, flashing 'I/O' LED.
If error code 6 is indicated, followed by error argument 4 by the "I/O" LED after run-up of the controller with red flashing of error code six times, followed by red flashing of the error argument four times, this indicates that an IP address has not yet been assigned.

3.1.7.1.2 Assigning of IP Addresses to Fieldbus Nodes

The following provides examples of fieldbus node IP address allocation using the "WAGO ETHERNET Settings" Program.

Note
The program "WAGO-ETHERNET Settings" is available for downloading at http://www.wago.com under: Downloads → AUTOMATION.
The program is also included on the CD "AUTOMATION Tools and Docs" (Item No.: 0888-0412) available from WAGO.

1. Start the **WAGO-ETHERNET Settings** program.

2. Click on **Identify** to identify the controller connected to the system.

3. Select the tab TCP/IP (see Fig. 3-2)

Fig. 3-2: Setting IP addresses using WAGO ETHERNET Settings g084906d

4. Enter the desired **IP Address** and, if applicable, the address of the subnet mask and gateway.

5. Click the button **Write** to transfer this address to the controller.

6. Click on **Call WBM**. An Explorer then opens with the Web-based management system in which you can make all further settings (see section 3.1.8.7).

3.1.7.1.3 Testing for Proper Functioning of the Fieldbus Node

1. Set up a (non-serial) link between the client PC and the controller to test communication with the controller and correct assignment of the IP address. The client PC must be equipped with a network card for this.

2. Call up the DOS prompt window: **Start / Programs / DOS prompt**.

3. Type the command **ping** using the IP address you have assigned, with the following syntax:

   ping [space] XXXX . XXXX . XXXX . XXXX

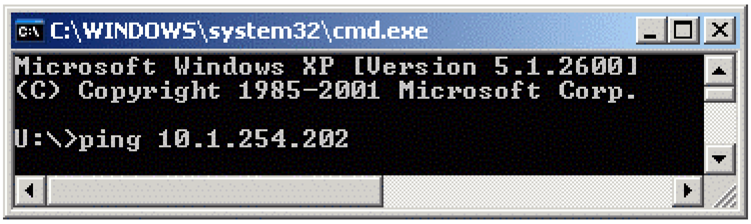


Fig. 3-1: Example for a fieldbus node function test G083070e

4. When you press the Return key the controller sends a reply to your client PC that is displayed in the DOS prompt window.
   Should the error message "Request timeout" appear, check your input against the IP address you assigned.

5. If the test is successful, you can close the DOS window. The node is now ready to communicate.

3.1.7.2 Commissioning with the WAGO BootP Server

An IP address and other parameters can be assigned to a coupler/controller in a TCP/IP network using the Bootstrap protocol (BootP). Subnet masks and gateways can also be transferred using this protocol.

Protocol communication comprises a client request and a server reply.

No IP address is available on commissioning of the controller. By default, the BootP protocol is activated in the controller. A broadcast request will be transmitted at Port 67 (BootP server) that contains the MAC address (MAC ID) for the controller.

The BootP server then receives this message. The server contains a database in which the MAC addresses and IP addresses are assigned to one another. When a MAC address is found a broadcast reply is transmitted via network.

The 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 controller. The controller recognizes by the MAC address that the message is intended for that particular controller and accepts the transmitted IP address into its network.

The following steps are included:

- Noting the MAC ID and setting up of the fieldbus nodes (see Section 3.1.7.2.1)
- Connecting of client PC and fieldbus nodes (see section 3.1.7.2.2)
- Determining IP addresses (see Section 3.1.7.2.3)
- Assigning IP addresses to the fieldbus nodes (see Section 3.1.7.2.4)
- Testing for proper functioning of the fieldbus nodes (see Section 3.1.7.2.5)
- Deactivating the BootP protocol (see Section 3.1.7.2.6)

3.1.7.2.1 Noting of the MAC ID and Set-Up of the Fieldbus Nodes

1. Before you set up the fieldbus node write down the MAC-ID of the BACnet/IP controller. The MAC-ID is applied to the back of the fieldbus controller, or on the self-adhesive peel-off strip on the side of the controller.

   MAC ID of fieldbus controller: 0 0 : 3 0 : D E : _ _ : _ _ : _ _

3.1.7.2.2 Connecting Client PC and Fieldbus Nodes

1. Connect the installed BACnet/IP controller to the client PC either directly, or using a 10BaseT or 100BaseTX cable via a hub. The controller transfer rate depends on the network data transfer rate of your client PC network card.

Note
If the fieldbus node is connected directly to the client PC, you will require a crossover cable instead of a straight-through cable (1:1).

2. Start the client PC that assumes the function of the master and BootP server.

3. Switch on the power at the controller (DC -24 V power supply unit).

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

If an error occurred during start-up, an error code is indicated by a red, flashing 'I/O' LED.
If error code 6 is indicated, followed by error argument 4 by the "I/O" LED after run-up of the controller with red flashing of error code six times, followed by red flashing of the error argument four times, this indicates that an IP address has not yet been assigned.

3.1.7.2.3 Determining of IP Addresses

If the client PC is already integrated into an IP network you can determine the client PC's IP address by performing the following steps:

1. On your screen desktop, go to **Start / Settings** and click **System control panel**.

2. Double click on the **Network** icon. The network dialog window then appears.

With Windows NT:

3. Select the tab **Protocols**

4. Mark the entry **TCP/IP protocol**

With Windows 2000/XP:

3. Select **Network- and Data transfer links**

4. In the dialog window that then appears, right click on **LAN** and open the link **Properties**.

5. Mark the entry **Internet protocol TCP/IP**

Note
If any of these entries are missing, install the required TCP/IP components and restart your PC. You must have the Windows NT installation CD, or the installation CD for Windows 2000/XP to install these components.

6. Then click the **Properties** button.
   The IP address, subnet mask and, where required, the client PC's gateway addresses are shown in the Properties window.

7. Be sure to note this data:
   Client PC IP address: _ _ _ _ . _ _ _ _ . _ _ _ _ . _ _ _ _
   Subnet mask: _ _ _ _ . _ _ _ _ . _ _ _ _ . _ _ _ _
   Gateway: _ _ _ _ . _ _ _ _ . _ _ _ _ . _ _ _ _

8. Now, select an IP address for your fieldbus node.

Note
The client at which the BootP server is executed must always have a fixed IP address. The controller and the client PC must be located in the same subnet.

9. Be sure to note the IP address you have selected:

   Fieldbus node IP address: _ _ _ _ . _ _ _ _ . _ _ _ _ . _ _ _ _

3.1.7.2.4 Assigning of IP Address to the Controller

The controller must be assigned an IP address before it can communicate properly.
This address can be assigned via "WAGO BootP server", or using a PFC program. When assigning an address using a PFC program, this can be done in WAGO-I/O-PRO CAA using the function block "Ethernet_Set_Network_Config" from the library "Ethernet.lib".

The following describes allocation of the fieldbus node IP address via the WAGO BootP server.

Note
The "WAGO-BootP Server" is available free of charge on the CD "AUTOMATION Tools and Docs" (Item No.: 0888-0412) or at our Website http://www.wago.com under Downloads → AUTOMATION.

Note
It is also possible to assign IP addresses under other operating systems (e.g. under Linux) as well as with other BootP servers.

Note
The IP address is assigned via straight-through cable, switches, hubs, or via direct link using a crossover cable. Addresses cannot be allocated via router.

3.1.7.2.4.1 BootP Table

The BootP table is the database for the BootP server. This table is available as a text file (bootptab.txt) on the client PC where the WAGO BootP server is installed.

Note
The WAGO BootP server must be installed correctly before the following steps can be performed:

1. On your PC, go to **Start** and select the menu item **Programs \ WAGO Software \ WAGO BootP Server**.

2. Click on **WAGO BootP server configuration**.

You are then provided with an editable table "bootptab.txt".

At the end of the list that highlights possible abbreviations that can be used in the BootP table, two examples are given, detailing the allocation of an IP address:

- **"Example of entry with no gateway"**
- **"Example of entry with gateway"**

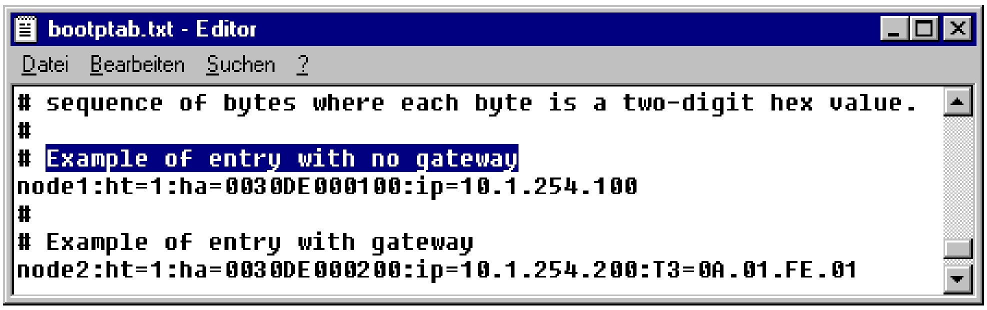


Fig. 3-12: BootP Table p012908d

The examples shown contain the following information:

Tab. 3-17: BootP Table Information

Information	Meaning
node1, node2	Any name for a node can be specified here.
ht=1	Here the hardware type of the network is specified. For ETHERNET the hardware type is 1. These numbers are explained in RFC1700.
ha=0030DE000100 ha=0030DE000200	Specify the hardware address (MAC ID) for the BACnet/IP controllers here (hexadecimal).
ip= 10.1.254.100 ip= 10.1.254.200	Specify the IP address for the BACnet/IP controller here (decimal)
T3=0A.01.FE.01	Specify the gateway address here (hexadecimal)
sm=255.255.0.0	The subnet mask for the subnetwork to which the BACnet/IP controller belongs can also be specified here (decimal).

The local network that is described in this description does not require a gateway. You can therefore apply the example "Example of entry with no gateway" here.

3. In the following text line, replace the 12-place hardware address located behind “ha=“ in this example.

```
node1:ht=1:ha=0030DE000100:ip=10.1.254.100
```

4. In place of this, enter the MAC address for your own controller.

5. If you would like to specify a name for your fieldbus node, delete "node1" in the text and enter the node name you wish to use.

```
node1:ht=1:ha=0030DE000100:ip=10.1.254.100
```

6. To assign the controller a specific IP address, mark the IP address given here in the example after
“ip=“ and enter your own IP address.

```
node1:ht=1:ha=0030DE000100:ip=10.1.254.100
```

7. Since you do not need the second example **"Example of entry with gateway" here, place the number sign** (#) in front of the text line in Example 2 as a comment symbol:

```
# node2:hat=1:ha=003 0DE 0002 00:ip=10.1.254.200:
                                        T3=0A.01.FE.01
```

This line will not be evaluated after this.

Note
To address additional fieldbus nodes, enter a similar text line for each node, with your own specific data.

8. In the menu **File** select the menu item **Save** to store the changed settings in the "bootptab.txt" file.

9. Close the editor.

3.1.7.2.4.2 BootP Server

1. On your PC, go to **Start** and select the menu item **Programs \ WAGO Software \ WAGO BootP Server**.

2. Click on WAGO BootP server to open the dialog window.

3. Click on Start in the dialog window that then appears. This activates the query/response mechanism of the BootP protocol.

A number of messages are then output at the BootP server. Error messages indicate that some services (such as Port 67, Port 68) are not defined in the operating system.

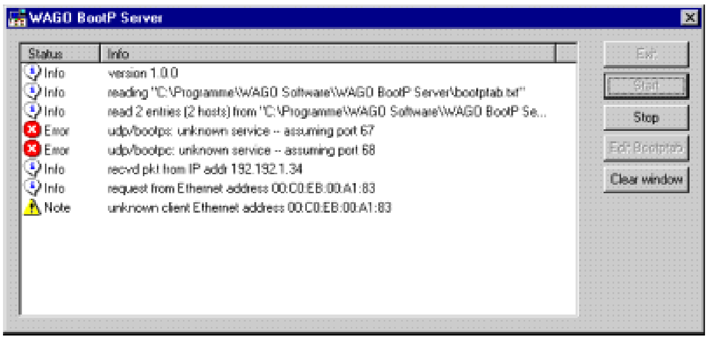


Fig. 3-13: Dialog window for the WAGO-BootP Server, with messages P012909d

4. To close the BootP server again, click **Stop** and then click the button **Exit**.

3.1.7.2.5 Testing the Function of the Fieldbus Node

1. In order to check communication with the controller and for correct IP address assignment, start the DOS prompt via **Start / Programs / Command prompt**.

2. Type the command **ping** using the IP address you have assigned, with the following syntax:

   ping [space] XXXX . XXXX . XXXX . XXXX

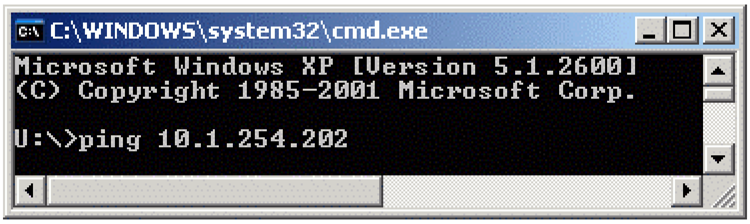


Fig. 3-2: Example for a fieldbus node function test G083070e

3. Press **Return**

The controller sends a reply to your client PC that is displayed in the DOS prompt window.
Should the error message "Request timeout" appear, check your input against the IP address you assigned.

4. If the test is successful, you can close the DOS command prompt window.

The node is now ready to communicate.

3.1.7.2.5.1 Reasons for Failed IP Address Assignment

- The controller MAC address does not correspond to the entry given in the "bootstrap.txt" file.

- The client PC on which the BootP server is running is not located in the same subnet as the controller; i.e., the IP addresses do not match
  Example: Client IP: 168.192.0.10 and controller IP: 10.1.254.5

- Client PC and/or controller is/are not linked to the ETHERNET.

- Poor signal quality (use switches or hubs)

3.1.7.2.6 Deactivating the BootP Protocol

By default, the BootP protocol is activated in the controller. When the BootP protocol is activated, the controller expects the BootP server to be permanently available. If there is no BootP server available after a PowerOn reset, the network will remain inactive.

You must deactivate the BootP protocol and set a fixed IP address. After that, a BootP server is no longer necessary.

Deactivation of the BootP protocol is performed using the HTML pages stored in the controller (see also Section 3.1.8.7).

Note
If the BootP protocol is deactivated after addresses have been assigned, the stored IP address is retained, even after an extended loss of power, or when the controller is removed.

1. Open the Web browser on your client PC (such as the Microsoft Internet Explorer) to have the HTML pages displayed.

2. Enter the IP address for your fieldbus node in the address line of the browser and press Return.

A dialog window then appears with a password prompt.

This is provided for secure access and entails three different user groups: "admin", "guest" and "user".

3. As Administrator, enter the user name: **"admin"** and the password **"wago"**.

A start page is then displayed in the browser window with information about your fieldbus controller. You can navigate to other information using the hyperlinks in the left navigation bar.

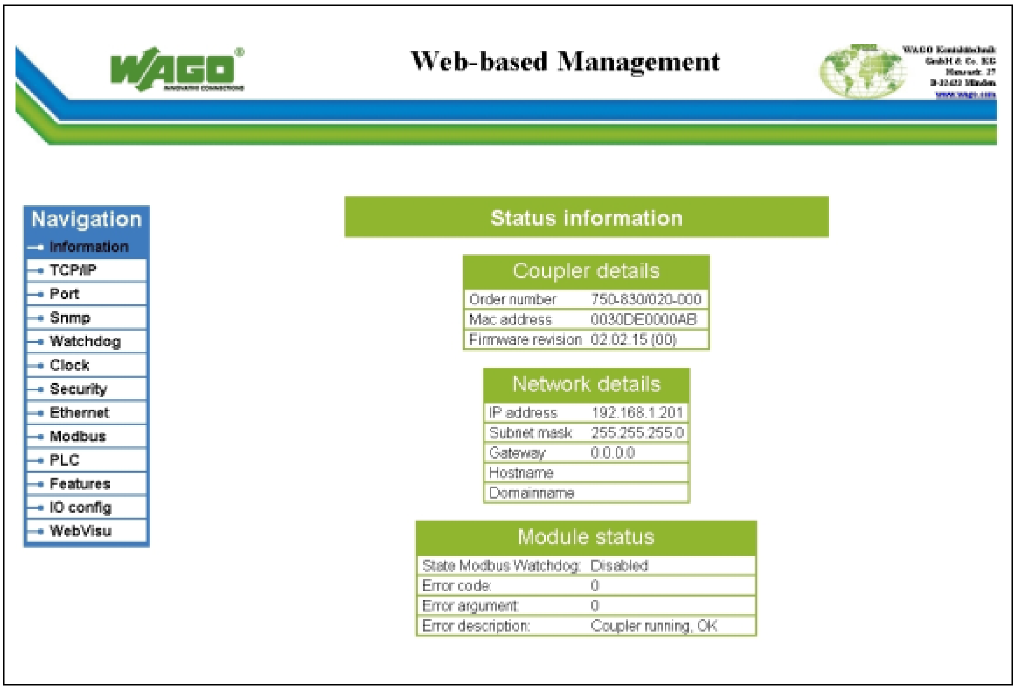


Fig. 3-2: HTML pages of the Web-based management system G083050e

Note
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 is to be used for the node IP address.

Note
If DHCP and BootP are activated and an ISDN/DSL router is incorporated in the network (factory default settings with DHCP server), addresses will be assigned automatically after a loss of power (loss of 24-V-DC power to controller) from the address range for the ISDN/DSL router. As a result, all controllers will be assigned new IP addresses!

4. In the left navigation bar click on **Port** to open the HTML page for selecting a protocol.

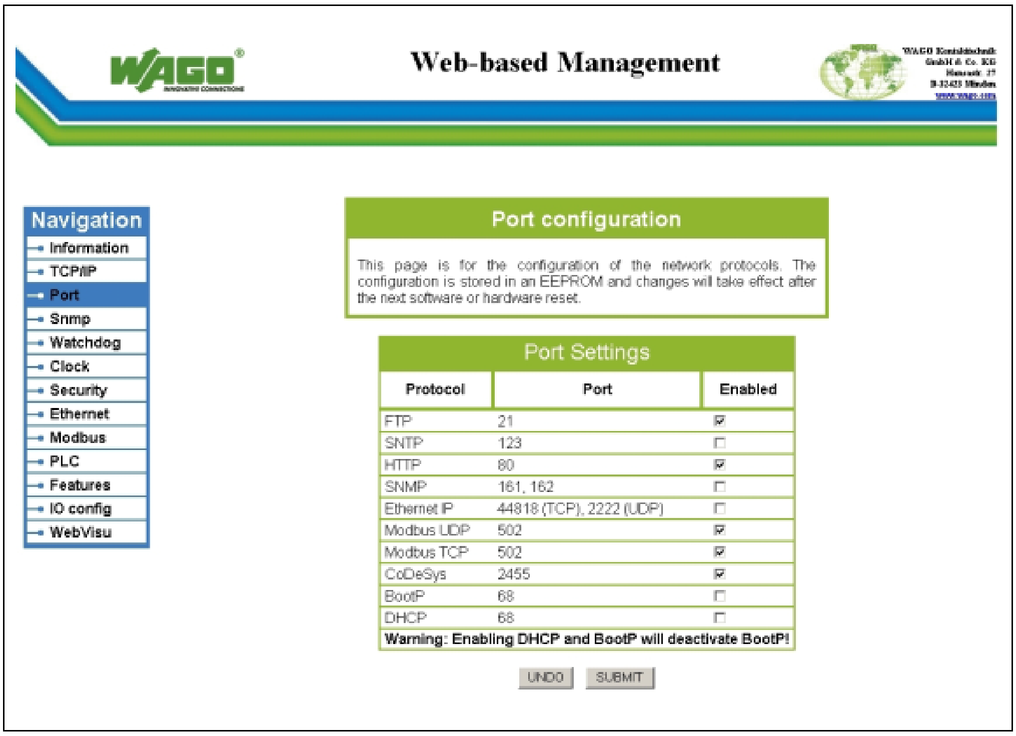


Fig. 3-3: Port configuration G083052e

You are shown a list of all the protocols supported by the controller. By default, the BootP protocol is activated in the controller.

5. Click the box behind **BootP** to remove the check mark. You have now deactivated the protocol.

You can also deactivate any other protocols that you no longer need in the same manner, or select desired protocols and activate them explicitly.

Since communication for each protocol takes place via different ports, you can have several protocols activated simultaneously; communication takes place via these protocols.

6. Click on SUBMIT and then switch off the power to the controller (hardware reset), or press down the mode selector switch.

7. Proceed according to section 3.1.7.1.2 and assign a fixed IP address to the controller.

The protocol settings are then saved and the controller is ready for operation.

If you have activated the MODBUS/TCP protocol, for example, you can now select and execute required MODBUS functions using the MODBUS master too, such as querying of the module configuration via register 0x2030.

If you have activated the WAGO-I/O-PRO for example, you can also program the controller via ETHERNET link using WAGO-I/O-PRO CAA in line with Standard IEC 61131-3.

3.1.8 Programming the PFC Using WAGO-I/O-*PRO* CAA

Using IEC 61131-3 programming, the 750-830 BACnet/IP Controller 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 CAA.

Note
IEC 61131-3 programming of the controller via ETHERNET requires that the check box "CoDeSys" be activated at the Website "Port Configuration" (see Section 1.1.8.7).
You can, however, also connect the client PC and controller serially for programming using a programming cable.

A description of programming using WAGO-I/O-PRO CAA is not included in this manual. The following sections, on the other hand, contain important information about creating projects in WAGO-I/O-PRO CAA and about special modules that you can use explicitly for programming of BACnet/IP controller. Explanations are also provided as to how the IEC 61131-3 program is transferred and how suitable communication drivers are loaded.

Additional Information
For a detailed description of using the software, refer to the manual for the "WAGO-I/O-*PRO* CAA". This manual is located at http://www.wago.com under: Documentation → WAGO-I/O-SYSTEM759 → WAGO-I/O-*PRO* → 759-333

1. Start the programming tool at **Start \ Programs \ WAGO-I/O-PRO** and **WAGO-I/O-*PRO* CAA**.

A dialog window then appears on which you can set the target system for programming.

Fig. 3-4: Dialog window for target system settings g083005e

2. Select the WAGO 750-830 BACnet/IP Controller by entering WAGO_750-830 and then click OK.

3. Create a new project under **File** / **New**.

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 bus module data properly in your new project, first compile the bus 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.

3.1.8.1 Configuration Using the WAGO I/O Configurator

The I/O Configurator is a plug-in incorporated into WAGO-I/O-PRO CAA for determining addresses for modules at a controller.

1. In the left half of the screen for the WAGO-I/O-*PRO*-CAA interface, select the tab **Resources**.

2. In the tree structure click **PLC configuration**. The I/O Configurator then starts up.

3. Expand the branch **Hardware configuration** in the tree structure with the sub-branch **K-Bus**.

4. Right click on K Bus or on an I/O module to open the menu for adding and attaching I/O modules.

5. Click on **Append Subelement** in the context menu. You can now select the desired I/O module from the I/O module catalog and attach it to the end of the K-bus structure using Insert and OK. In this case, the command "Insert element" is deactivated.

6. To insert an I/O module in front of a selected I/O module in the K Bus structure, right click on an I/O module that has already been selected and then click Insert element. In this case, the command "Insert sub-element" is deactivated.

You can also access these commands with the "Insert" menu in the main window menu bar. The dialog window "I/O configuration" for selecting modules is opened both by "Attach sub-element" and by "Insert element." In this dialog window, you can position all the required modules in your node configuration.

7. Position all of the required I/O modules until this arrangement corresponds to the configuration of the physical node. Complete the tree structure in this process for each module in your hardware that sends or receives data, either bit-by-bit or word-by-word (data width/bit width > 0).

Note
The number of modules that send or receive data must correspond to the existing hardware (except for supply modules, copying modules or end modules, for example). The number of input and output bits or bytes of the individually connected bus modules can be found in the corresponding descriptions of the bus modules.

Additional Information:
To obtain further information about an I/O module, either select that module from the catalog, or in the current configuration and then click the button **Data Sheet**. The module is then shown in a separate window with its associated data sheet.

Note
For the current version of the data sheets go to 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 updated.

If required, you can also modify the authorization privileges for individual 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 on a module in the configuration.

10. In the right dialog window under the tab
"Module parameters" define for each module from where access to the module data is to be carried out.

You can choose from the following settings in the column "Value" for this:

- PLC (standard setting) - Access from PFC
- MODBUS TCP/UDP - Access from MODBUS/TCP
- MODBUS RTU - Access from MODBUS/RTU
- BACnet - Access from BACnet/IP

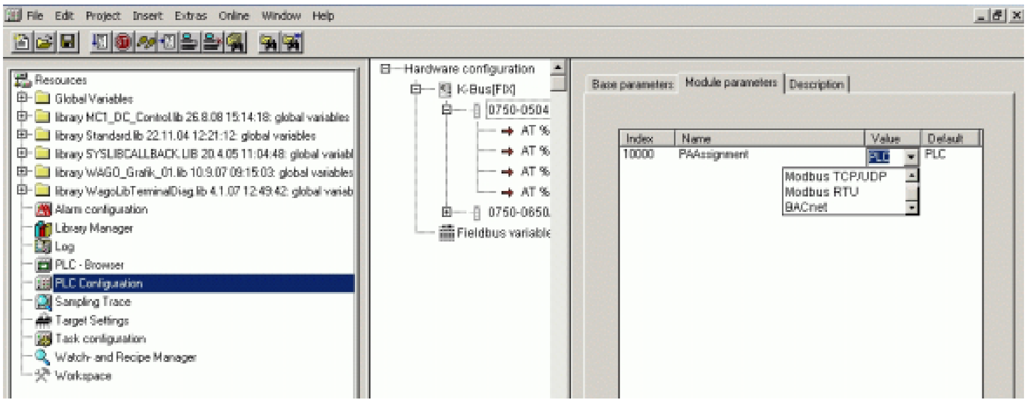


Fig. 3-5: Write access over module parameters g083023e

After completing these settings you can begin with IEC 61131-3 programming. The "EA-config.xml" configuration file is generated as soon as the project has been transferred.

Additional Information:
For a detailed description of how to use the WAGO-I/O-*PRO* CAA software and the I/O Configurator, refer to the online help function for WAGO-I/O-*PRO* CAA.

Note
You can also create the file "EA-config.xml" using an editor and store it in the controller directory "/etc" by means of FTP.
Configuration using the file "EA-config.xml" that is already stored in the controller is described in the following section.

3.1.8.1.1.1 Configuration Using the "EA-config.xml" File

Note
If you wish to perform module assignment directly using the "EA-config.xml" file stored in the controller, do not save any configuration data in WAGO-I/O-PRO CAA prior to this, as the file is overwritten by entries in the WAGO-I/O-PRO CAA on each download.

1. Open any FTP client. You can also use the Windows FTP client in the DOS prompt window.

2. Type in the IP address of the controller to access the controller file system.

3. Then, enter **admin** as the user login and **wago** as the password.

4. The file "EA-config.xml" is located in the "etc" folder. 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:

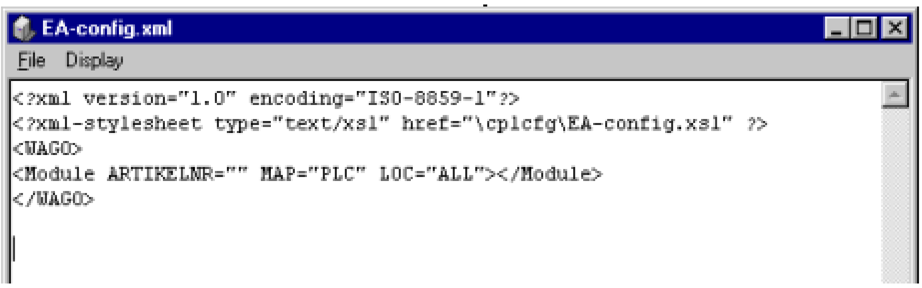


Fig. 3-6: EA-config.xml P012913x

The fourth line contains the necessary information for the first bus module. The entry MAP="PLC" assigns write access privileges to the IEC 61131-3 program for the first module.
If you wish to change the access rights, replace "PL" with "FB3" as the access privileges from BACnet.

5. Then complete the fourth line for each individual module using this syntax and set the corresponding assigned access privileges:

   <Module ITEMNO="" MAP="(e.g..) PLC" LOC="ALL"> </Module>

Note
It is imperative that the number of line entries matches the number of existing hardware modules.

6. Save the file and reload it to the controller file system via the FTP client.

You can then begin with IEC 61131-3 programming.

Additional Information
For a detailed description of how to use the software, please refer to the WAGO-I/O-PRO CAA manual. The manual available at http://www.wago.com under Documentation → WAGO-I/O-SYSTEM 759 → WAGO-I/O-*PRO* → 759-333

3.1.8.2 Creating/Exporting the SYM_XML File

Note
If you are in the simulation mode you cannot 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 CAA 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**. This check mark must appear in black!

Once these options have been activated, an SYM_XML file will be generated automatically with project variables when a project is compiled.

Note
If the check mark in the box **Issue object variables** is gray, this means that it has not been explicitly set. Click in the box again to make sure that the check mark appears in black.

3.1.8.3 ETHERNET Libraries for WAGO-I/O-PRO CAA

Various libraries are available in WAGO-I/O-*PRO* CAA for different IEC 61131-3 programming tasks. These contain modules for universal use and can, thereby, facilitate and speed up the creation of your program.

Additional Information
All libraries are included on the installation CD for the software WAGO-I/O-*PRO* CAA in the folder directory:
CoDeSys V2.3\Targets\WAGO\Libraries\...

Some libraries, such as 'standard.lib' and 'IECsfc.lib' are normally incorporated; the ones described below, however, are specific to ETHERNET projects with WAGO-I/O-*PRO* CAA:

Tab. 3-18: ETHERNET libraries for WAGO-I/O-*PRO* CAA

Library	Contents
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 PC via TCP protocol to exchange data with any potential UDP server or client PC via UDP protocol
WAGOLibModbus_IP_01.lib	Function blocks that can set up links with one or more slaves
ModbusEthernet_03.lib	Function blocks that enable data exchange with several MODBUS slaves
ModbusEthernet_04.lib	Function blocks for data exchange with several MODBUS/TCP/UDP slaves Also a function block that provides a MODBUS server that maps the MODBUS services on a word array.
SysLibSockets.lib	Function block for access to sockets for communication via TCP/IP and UDP.
WagoLibSockets.lib	Function blocks for access to sockets for communication via TCP/IP and UDP Contains additional functions in addition to SysLibSockets.lib
Mail_02.lib	Function block for sending e-mails
WAGOLibMail_01.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)

These libraries are included on the WAGO-I/O-PRO CAA CD.
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.

Additional Information
For a detailed description of the function blocks and use of the software, refer to the WAGO-I/O-PRO CAA manual at http://www.wago.com under: Documentation →
WAGO-I/O-SYSTEM 759 → WAGO-I/O-*PRO* → 759-333
or the online Help function for WAGO-I/O-*PRO* CAA.

3.1.8.4 General Information about IEC Tasks

Note
Please note the following information when programming your IEC tasks.

- IEC tasks must have different priorities, as otherwise an error will occur during translating of the application program.

- 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.

- 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!

- Running tasks are halted after each task cycle for a duration that is half the time that the task requires (minimum 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

- If no task is incorporated in the task configuration, a freely running default task is created internally during the translation. The watchdog for this task is deactivated. 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.

- Sensitivity is of significance only for cyclic 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:
  If the set, maximum runtime is less than or equal to the call interval, a violation of the call interval likewise results in the watchdog event being triggered, regardless of the value set for sensitivity.

- If the set runtime is greater than the call interval the watchdog event is triggered when the maximum runtime is reached, regardless of the value set for sensitivity.

3.1.8.4.1 IEC Task Sequence

1. Determine the system time (tStart).

2. If no full internal bus cycle has run since the last time the outputs were written:
→ Wait until the next internal bus cycle is completed.

3. Reading of inputs and reading back of the outputs from the process image.

4. If the application program has been started.
→ Execute the program codes for this task.

5. Writing of the outputs to the process image.

6. Determine the system time (tEnd).
→ tEnd - tStart = runtime for the IEC task

3.1.8.4.2 Overview of Most Important Task Priorities

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 tasks 1-10)
IEC tasks with this priority may be interrupted by the internal bus tasks. Therefore, configuration for the connected modules and communication via fieldbus with the watchdog activated for the task call interval must be taken into account here.

PLC-Comm task (internal)
The PLC-Comm task is active when logged in and takes up communication with the WAGO-I/O-PRO CAA gateway.

Background task (IEC tasks 11-31)
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.

Tab. 3-19: Task priorities

Priority	Task
0 (high)	Internal bus task, fieldbus task
1	Normal task
2	PLC-Comm task
3 (low)	Background task

Definition: Processes with the highest priority are identified by the lowest numbers. These processes are handled by all other processes.

Additional Information
For a detailed description of the programming tool WAGO-I/O-*PRO* CAA refer to the manual WAGO-I/O-*PRO* CAA at http://www.wago.com under: Documentation → WAGO-I/O-SYSTEM759 → WAGO-I/O-PRO → 759-333

3.1.8.5 System Events

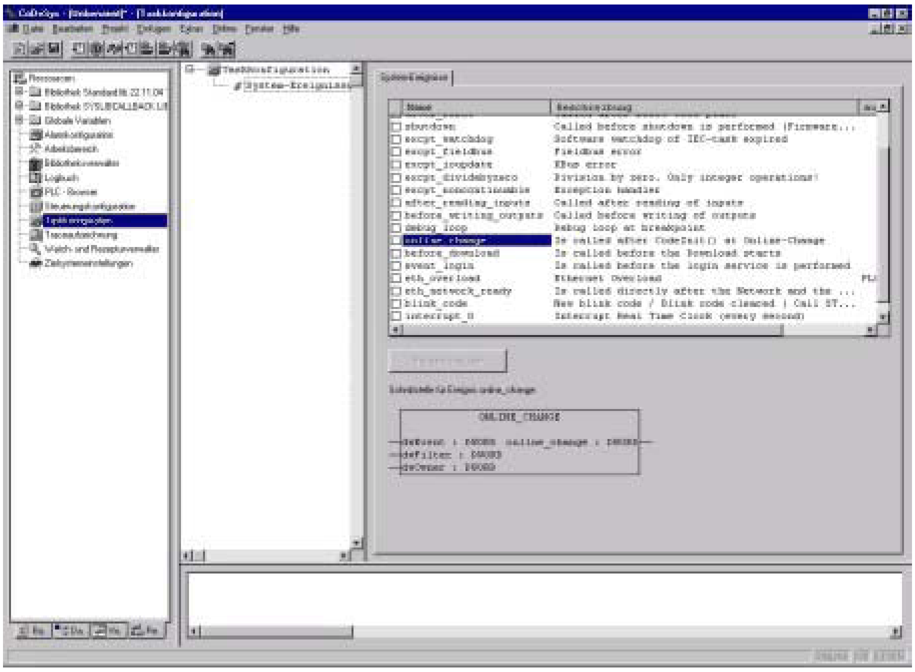


Fig. 3-7: System events p912277d

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 CAA / "Resources" tab" / "Task configuration" / "System events".
A module is called up by the event when the corresponding item is activated; i.e., when there is a check in the check box in the first column. Activation or deactivation is done with a mouse click on the check box.

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* CAA at http://www.wago.com under:
Documentation → WAGO-I/O-SYSTEM 759 → WAGO-I/O-*PRO* → 759-333

3.1.8.6 Transfer of IEC 61131-3 Program

Transfer from the PC to the controller of the program for the created IEC 61131-3 application can be performed two ways:

- Direct transfer via serial RS232 port
- Transfer by means of TCP/IP via fieldbus

Suitable communication drivers are required for transfer; these can be configured using WAGO-I/O-PRO CAA under Online / Communication parameters.

Note
When selecting an appropriate driver, ensure that the communication parameters are set and matched accordingly.

3.1.8.6.1 Transfer via Serial Service Port

Use the WAGO communication cable to set up a physical connection via serial service port. This is included in the scope of supply for the programming software WAGO-I/O-PRO CAA (Item No.: 759-333) or can be procured as an accessory item under order no. 750-920.

Notice
The communication cable 750-920 may not be connected or removed when the system is energized; i.e., there must be no power to the controller!

Use the WAGO communication cable to connect the COMX port of your PC to the controller communication port.
A communication driver is required for serial data transfer. This driver and its parameters must be entered in the WAGO-I/O-*PRO* CAA in the dialog window "**Communication parameters**".

1. Start the WAGO-I/O-PRO CAA software under **Start \ Programs \ WAGO-I/O-PRO** and **WAGO-I/O-*PRO* CAA**.

2. In the menu **Online select the item Communication parameters**

The dialog window "Communication parameters" then appears. This window is empty in its default settings.

3. Click **New** to set up a link and then enter a name, such as RS232.

4. In the selection window, mark the required driver in the right side of the window, **Serial (RS232) – 3S Serial RS232 driver**, to configure the serial link between the PC and the controller.

The following standard entries are shown in the center dialog window:

- Port: COM1
- Baud rate: 19200
- Parity: Even
- Stop-bits: 1
- Motorola byteorder: No

If necessary, change the entries accordingly by clicking on the respective value and editing it.

5. Confirm these settings by clicking **OK**

The RS232 port is now configured for transferring the application.

Additional information
For details on installing the communication drivers and using the software, refer to the WAGO-I/O-*PRO* CAA. The manual available at http://www.wago.com under: Documentation → WAGO-I/O-SYSTEM 759 → WAGO-I/O-*PRO* → 759-333

Note
The controller mode selector switch must be at either the center or top position to access the controller.

6. Under **Online**, click the menu item **Login** to log in to the controller.
   (The WAGO-I/O-*PRO* CAA Server is active during online operation. The communication parameters cannot be called up during this time.)

If there is no program in the controller, a window appears asking whether or not the program is to be loaded.

7. Respond with **Yes** to load the current program.

8. Once the program has been loaded, start program processing in the menu **Online**, menu item **Start**.
   "ONLINE" and "RUNNING" will then appear at the right of the status bar.

9. To terminate online operation, click the menu item **Log off** in the **Online** menu.

3.1.8.6.2 Transfer via Fieldbus

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 CAA in the dialog window "Communication parameters".

1. Start the WAGO-I/O-PRO CAA software under Start / Programs / WAGO-I/O-PRO and WAGO-I/O-PRO CAA, or by clicking the program icon on the desktop.

2. In the menu **Online** select the item **Communication parameters**.

The dialog window "Communication parameters" then appears. This window is empty in its default settings.

3. Click **New** to set up a connection and then specify a name.

4. In the selection window mark the desired driver on the right-hand dialog side (e.g. TCP/IP), to configure the serial connection between PC and the controller).

The following standard entries are shown in the center dialog window:

Address:	the IP address of the controller
Port:	2455
Motorola byteorder:	No

Change any entries as you may require.
You have now configured the TCP/IP link with the communication parameters/drivers.

Note
The controller must have an IP address before it can be accessed. The controller mode selector switch must be set to the center or top position.

5. Under **Online**, click the menu item **Login** to log in to the controller.
(The WAGO-I/O-*PRO* CAA Server is active during online operation. The communication parameters cannot be called up during this time.)

If there is no program in the controller, a window now appears asking whether or not the program is to be loaded.

6. Respond with **Yes** to load the current program.

7. Once the program has been loaded, start program processing in the menu **Online**, menu item **Start**.
"ONLINE" and "RUNNING" will then appear at the right of the status bar.

8. To terminate online operation, click the menu item **Log off** in the **Online** menu.

3.1.8.7 The Web-Based Management System (WBMS)

HTML pages containing information and setting options are stored in the controller as referred to as the Web-based management system. Use the menu on the left to navigate through these pages.

Information

Click the link "Information" to view status information about your controller and network.

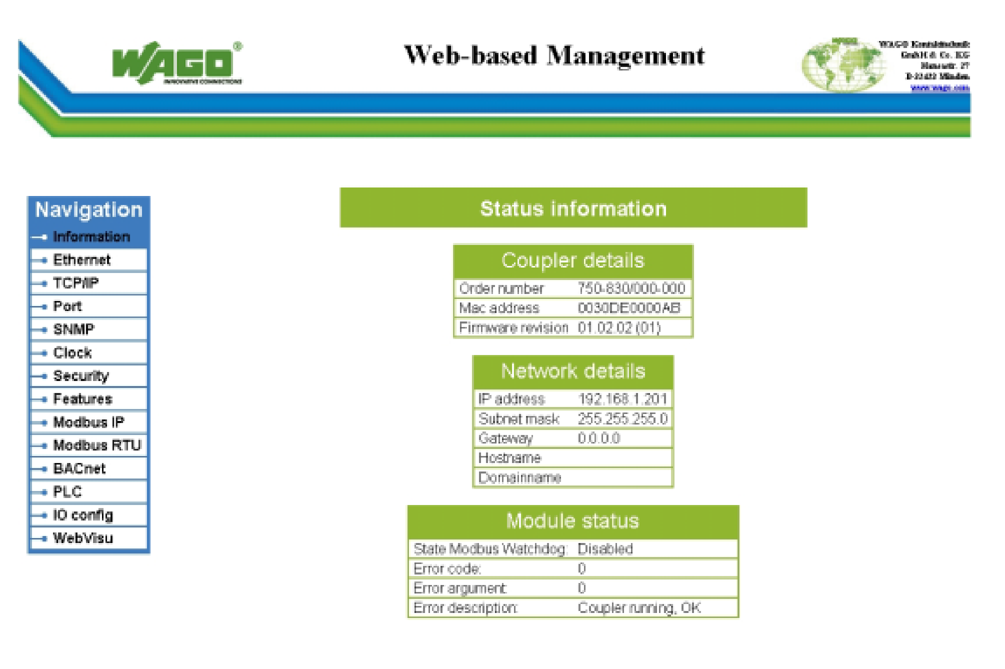


Fig. 3-8: Web-based Management System: Information

ETHERNET

Over the "Ethernet" link, you will reach a website on which you can configure the bandwidth limit and transmission rate for ETHERNET communication. With the BACnet/IP Controller, you will use Port 1 while setting the transmission rate ("Speed Configuration") (see Tab. 3-20).

Tab. 3-20: Set transmission rate ("Speed Configuration“)

Parameter	Description
Enable Port	Deactivates the ETHERNET port. Both ports cannot be deactivated at the same time.
Enable Power-Save Mode	Activates the energy saving mode for the ETHERNET port
Enable Auto MDI-X	Activates the Auto MDI-X function of the ETHERNET port. The "Auto MDI-X" function employs an internal switching functionality to allow the use of either crossed (Crossover) or straight-in (Patch) cables.
Enable Autonegotiation	Activates the autonegotiation function of the ETHERNET port. The "Autonegotiation" function enables automatic determining (negotiating) of the best-possible baud rate and the optimal transfer mode between two link partners.
10 MBit Half Duplex	Configures the ETHERNET port for a baud rate of 10 MBit, with a half duplex transfer mode.
10 MBit Full Duplex	Configures the ETHERNET port for a baud rate of 10 MBit, with a full duplex transfer mode.
100 MBit Half Duplex	Configures the ETHERNET port for a baud rate of 100 MBit, with a half duplex transfer mode.
100 MBit Full Duplex	Configures the ETHERNET port for a baud rate of 100 MBit, with a full duplex transfer mode.
Enable IEEE 802.3x Full Duplex Flow Control	Activates automatic flow control for the full duplex transfer mode

A bandwidth limit function can also be configured for the ETHERNET port. The following modes can be configured under Section "Bandwidth/Sniffer Configuration" in the option fields "Limit mode":

Mode	Description
ALL	All ETHERNET data packets are limited
BC MC FU	Broadcast, multicast and flooded-unicast data packets are limited
BC MC	Broadcast and multicast data packets are limited
BC	Broadcast data packets are limited

A set data transfer rate can be defined for the set mode with the option field "Input/Output Limit Rate." For this, port 3 is the internal ETHERNET port linked to the CPU. Bandwidth limiting configured for Port 3 will not have an effect on the data transfer of ETHERNET Port 1!.

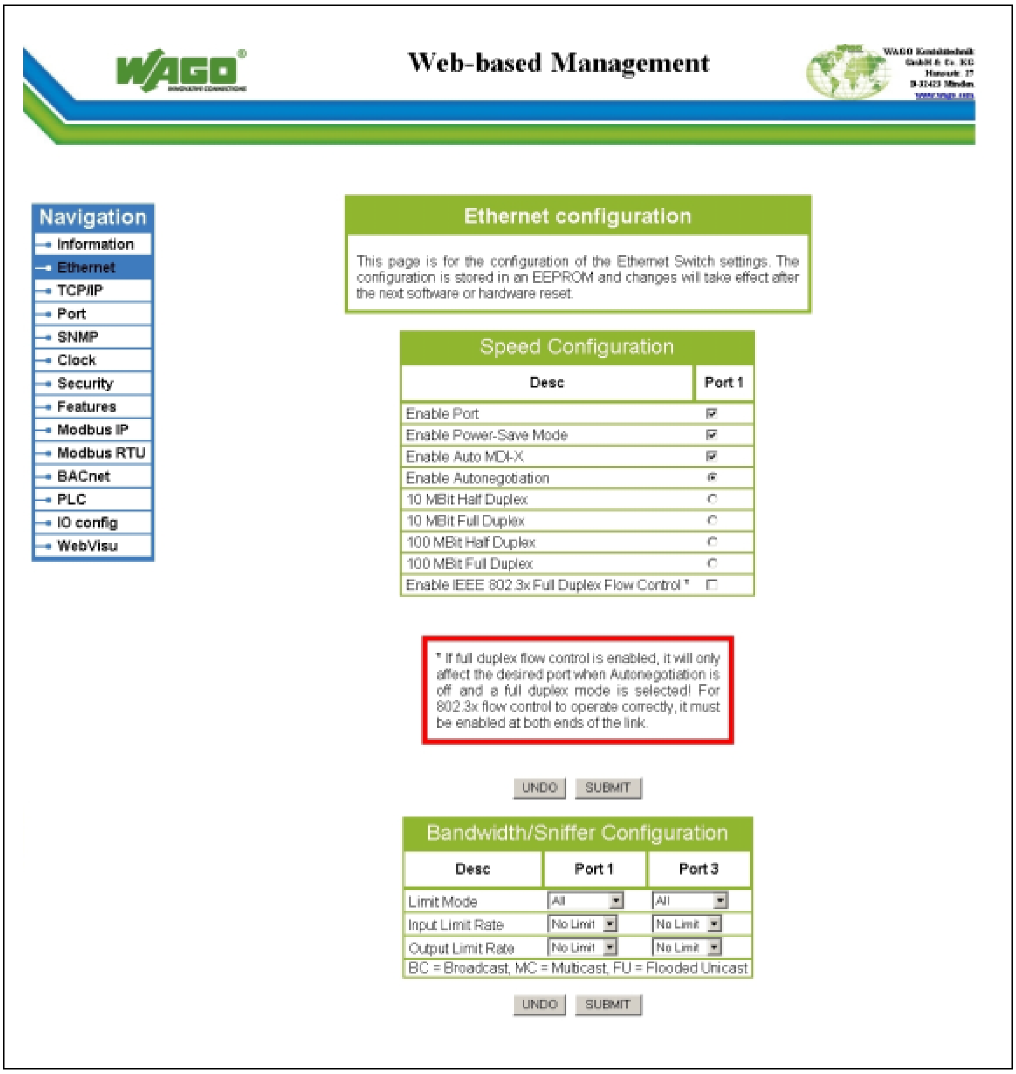


Fig. 3-9: Web-based Management System: ETHERNET

TCP/IP

Click the link "TCP/IP" to go to a Web site where you can specify the settings for the TCP/IP protocol. This protocol forms the basis for network data transfer.

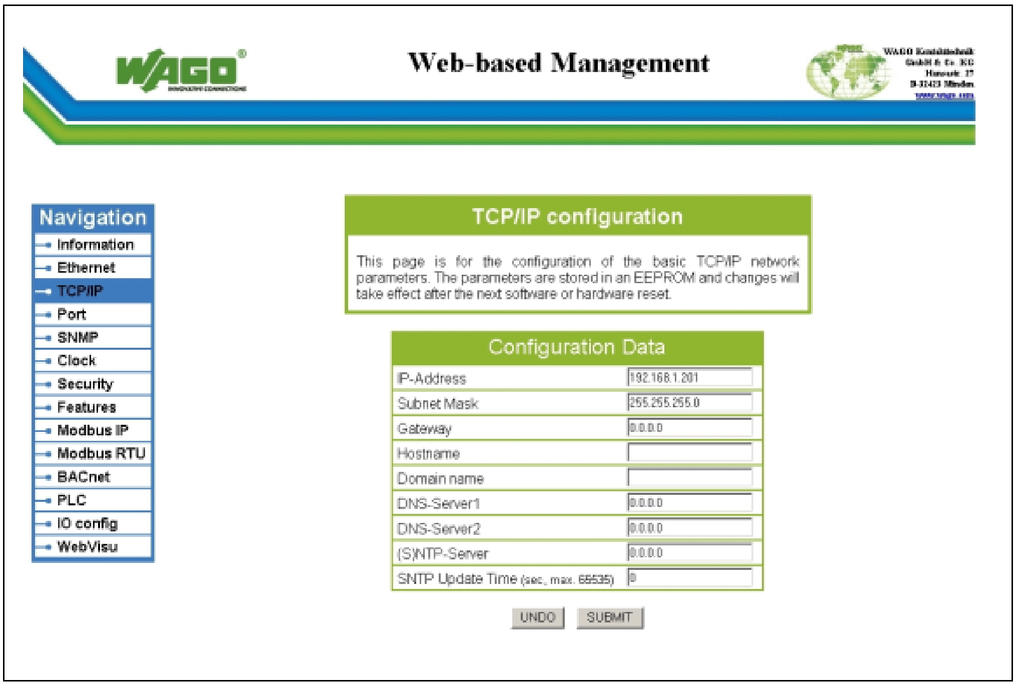


Fig. 3-10: Web-based Management System: TCP/IP

Port

Click the "Port" link to go to the "Port configuration" page, where you can activate or deactivate the desired protocol.
Normally, FTP, HTTP, MODBUS/UDP, MODBUS/TCP, WAGO Services, and CoDeSys are activated.

Fig. 3-11: Web-based Management System: Port

SNMP

Click the link "SNMP" to go to a Web site where you can specify the settings for the simple network management protocol. This protocol forms the basis for transfer of control data.

Fig. 3-12: Web-based Management System: SNMP

Clock

Click the link "Clock" to go to a Web site where you can specify the settings for the internal real-time clock.
Here, enter the current time and date and also select standard or daylight saving time.

Note
The internal clock must be (re)set on initial startup, or after 6 days without power. The "I/O" LED for the controller will flash with the error code 1/10 RTC-Powerfail if the clock is not set.

Note
Switch-over between standard and daylight saving time via Web-based management system is required when synchronizing the controllers in your network using a time server. The change-over is resolved via function block PrgDaylightSaving, which you must integrate into the WAGO-I/O-*PRO* CAA using the library DaylightSaving.lib. From that point, change-over will be performed automatically, allowing all functions to be executed properly and at the right time.

Note
Call up the Web-based management system and set the actual time under "Clock" in order to restore proper functioning of the controller in the event of an RTC-Powerfail. The controller will then be fully operational again.

Note
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
You can also use a WAGO-RTC module 750-640 at your node to utilize the actual time (encoded) in your higher-order control system. An even greater degree of accuracy is achieved using this "Real-Time Clock" module than that obtained using the real-time clock in the controller.

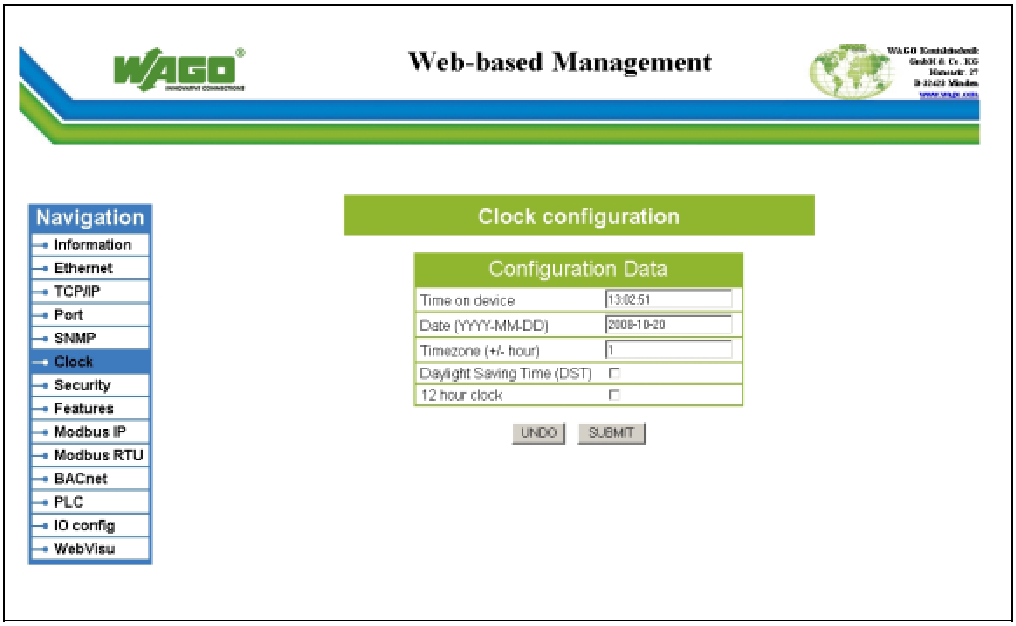


Fig. 3-13: Web-based Management System: Clock

Security

Click the "Security" link to go to a Web site at where you can configure read and/or write access privileges for various user groups using passwords to protect the configuration against unauthorized/inadvertent changes.

A distinction is drawn between the following user groups:

Tab. 3-21: User groups and their rights

User	Password	Authorization
admin	wago	Reading/writing values, triggering of software reset and access to security settings
user	user	Reading/writing values, initiation of a software reset, no editing of security settings
guest	guest	Read only

Note
The following restrictions apply to passwords: max.16 characters, only letters and digits, no special characters, symbols or umlauts.

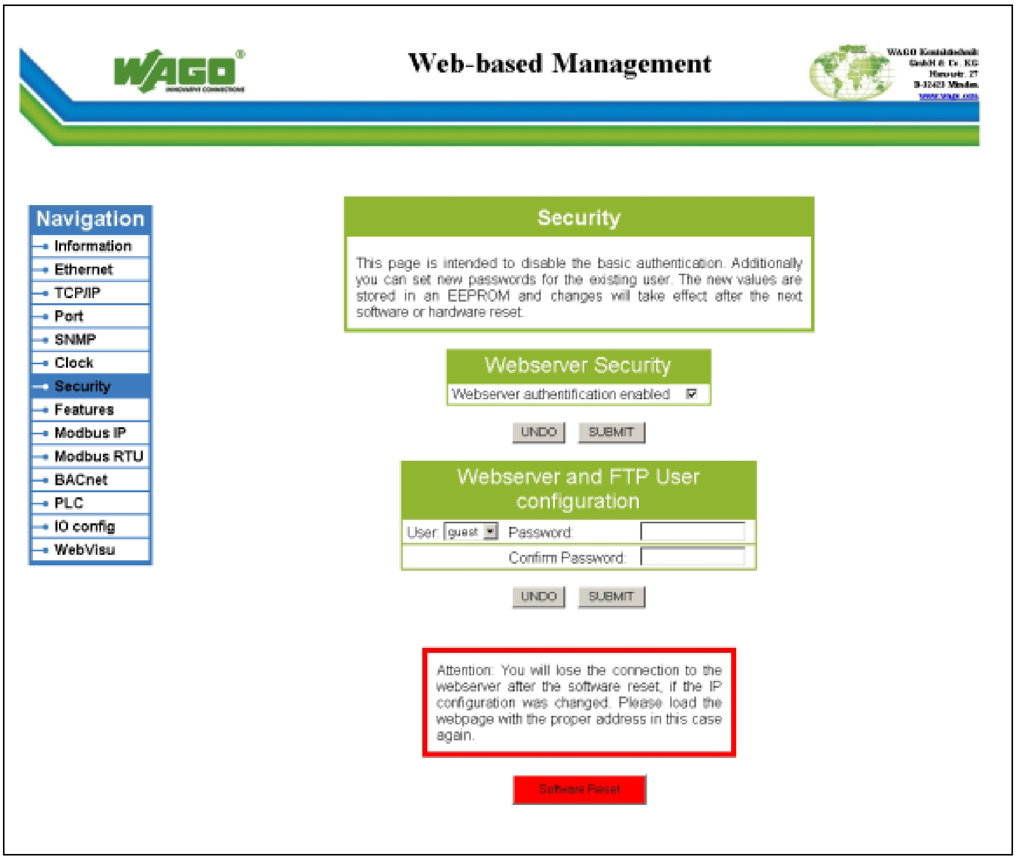


Fig. 3-14: Web-based Management System: Security

Features

Click the link "Features" to go to a Website at which you can activate or deactivate additional functions.

The "Autoreset on system error" function enables an automatic software reset to be conducted when a system error occurs.
This function 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 auto restart is executed as soon as the controller has an error status that requires a restart.

With the original factory settings, this function is deactivated by 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.

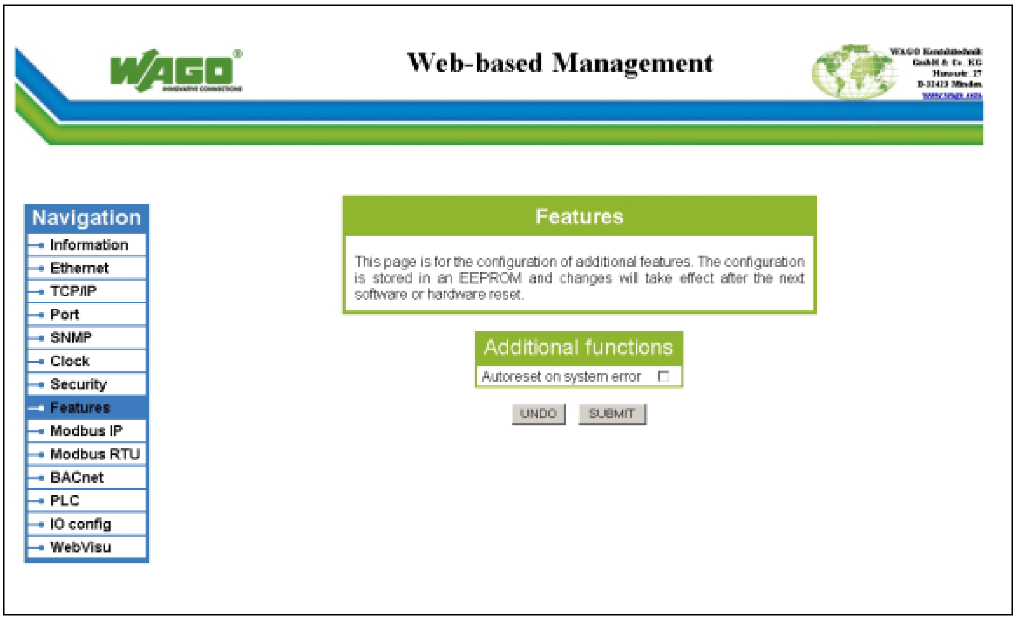


Fig. 3-15: Web-based Management System: Features

MODBUS IP

Click the link "Modbus IP" to go to a Web site where you can specify the settings for the MODBUS watchdog.

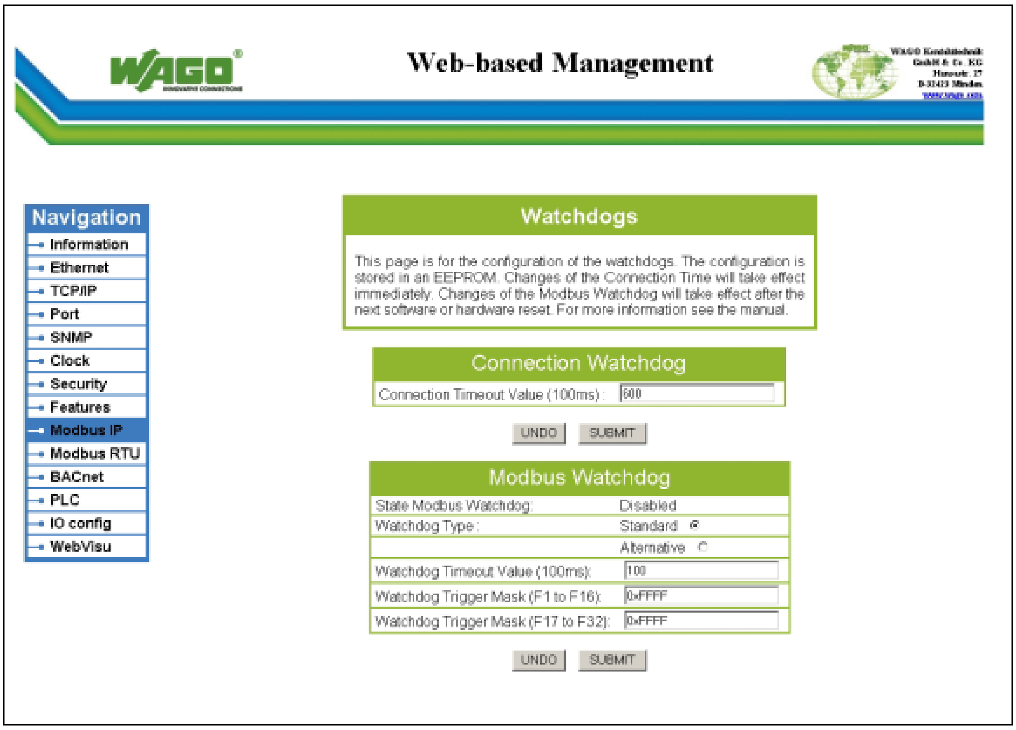


Fig. 3-16: Web-based Management System: MODBUS IP

MODBUS RTU

Click the link "MODBUS RTU" to go to a Web site where you can specify the settings for the MODBUS/RTU protocol.

On this page, you set the baud rate of 9600 (standard), 19200 or 57600. With each byte, a parity bit can also be sent. Errors in data transmission are detected with the aid of the parity bit. A differentiation is made between even (even parity), uneven (odd parity) and no parity testing (no parity).

Enter the "Slave Device Address" in the range 0-255 and select "Override default fieldbus settings" in order to assign all modules MODBUS/RTU by default instead of MODBUS/IP.

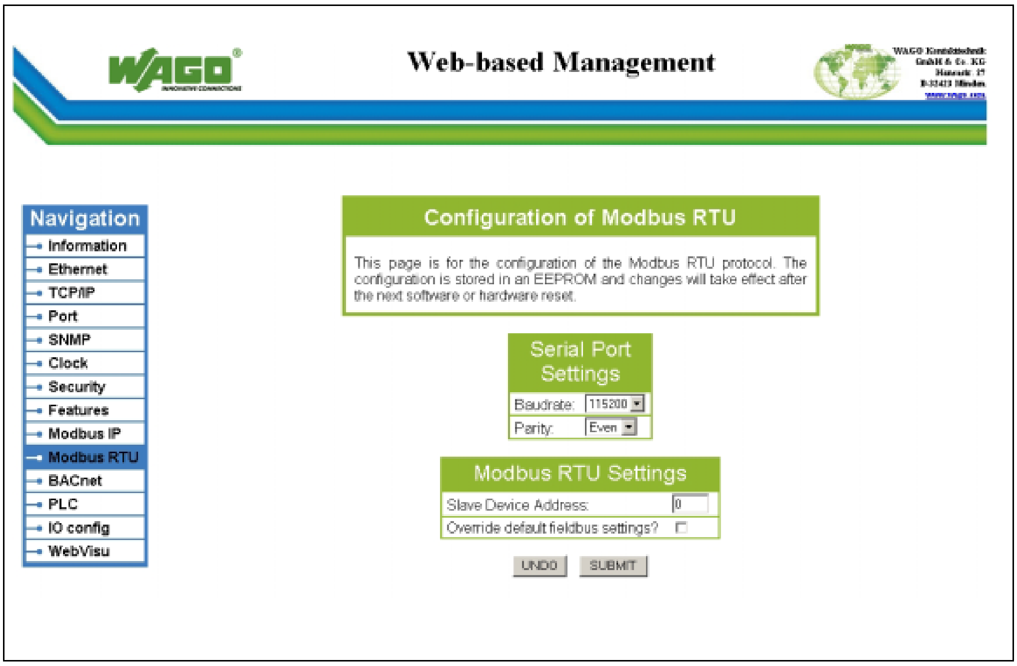


Fig. 3-17: Web-based Management System: MODBUS RTU

BACnet

You can set the transmission rate of the internal data bus and the UDP port on the "BACnet" page.

In the "UDP Port" field, enter the UDP port for BACnet/IP that is to be used.

If you place a check mark in the "Non-adaptive internal data bus transmission rate in ms" box (default setting), the transmission rate will be constant and will not be adapted to the node configuration.

This not only helps reduce the run-up time, but it also reduces the transmission rate. Remove the check mark to increase the transmission rate.

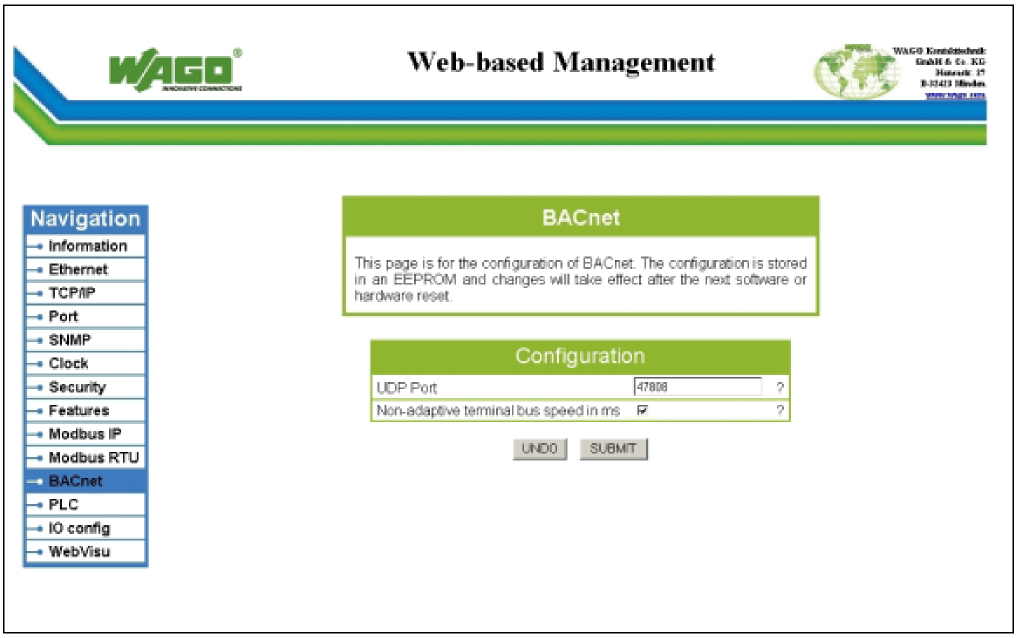


Abb. 3-18: Web-based Management System: BACnet

PLC

Click the "PLC" link to access a Web site where you can define the PFC functionality settings for your controller.

Use the function "Process image - Set outputs to zero if user program is stopped" to define the status of the outputs when your application program quits.
If there is a check in the box for this function, all outputs will be set to zero; if there is no check, the outputs will retain their current value.

Use the function "Default webpage" to define that the page "Webvisu.htm" be used as the starting page instead of the standard page "Status information" that is displayed when WMBS is accessed.

Note
The "Webvisu.htm" page does not have any hyperlinks to 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.

Use the "I/O configuration - Compatible handling for ea-config.xml" to define the procedure employed for writing privilege access to outputs. Here, note whether a control system configuration has already been created and, if so, whether this configuration is correct or not.

Handling for this is shown in the following table.

Control system configuration in the project	I/O configuration (function activated)	I/O configuration (function deactivated, standard setting)
none	Write 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 write 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	Write privilege to the module outputs is taken from the control system configuration. A corresponding ea-config.xml file is generated in the file system.	
faulted	The standard fieldbus is granted write privileges to the outputs of all the modules.	

When the function "I/O configuration – Insert monitoring entries into ea-config.xml" is activated, the current process values will also be shown for the data channels that are displayed on the "I/O config" Web page.

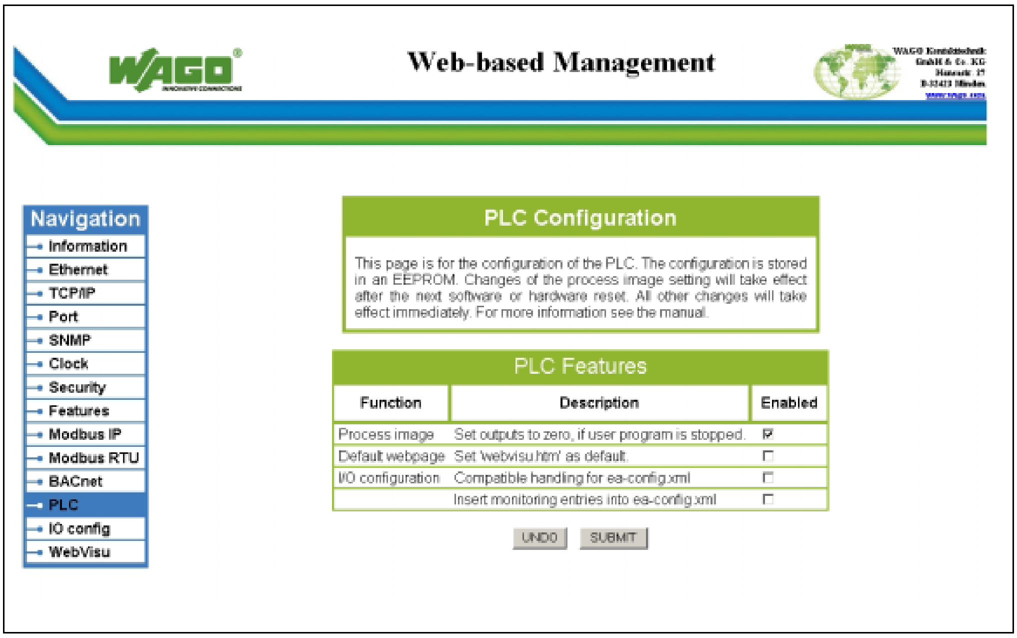


Fig. 3-19: Web-based Management System: PLC

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 CAA 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.

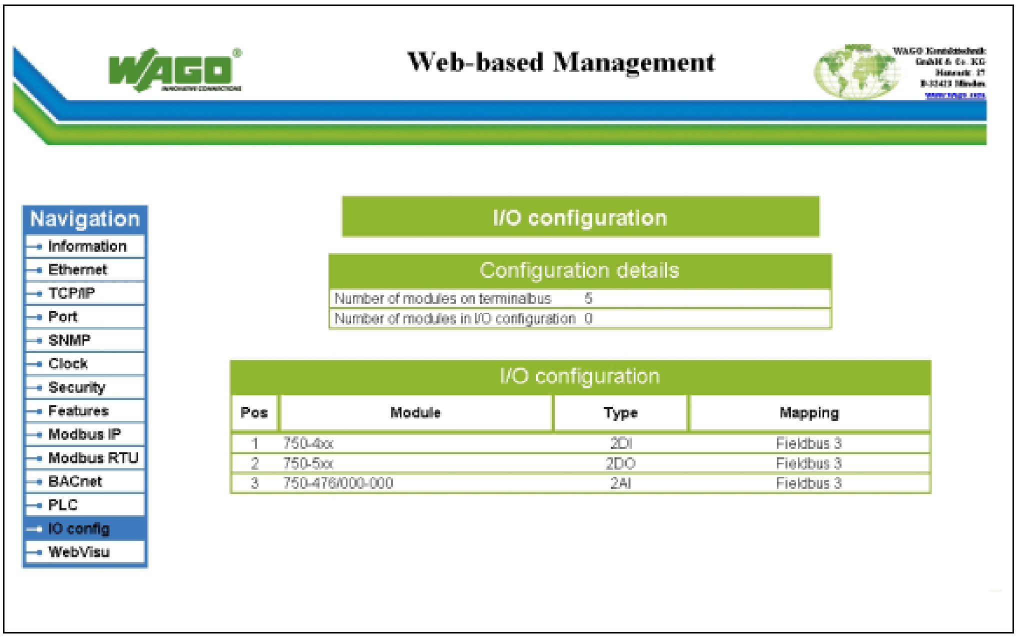


Fig. 3-20: Web-based Management System: I/O config

Additional Information
For more detailed information about the WAGO-I/O-*PRO* CAA I/O Configurator, refer to the Section 3.1.7"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.

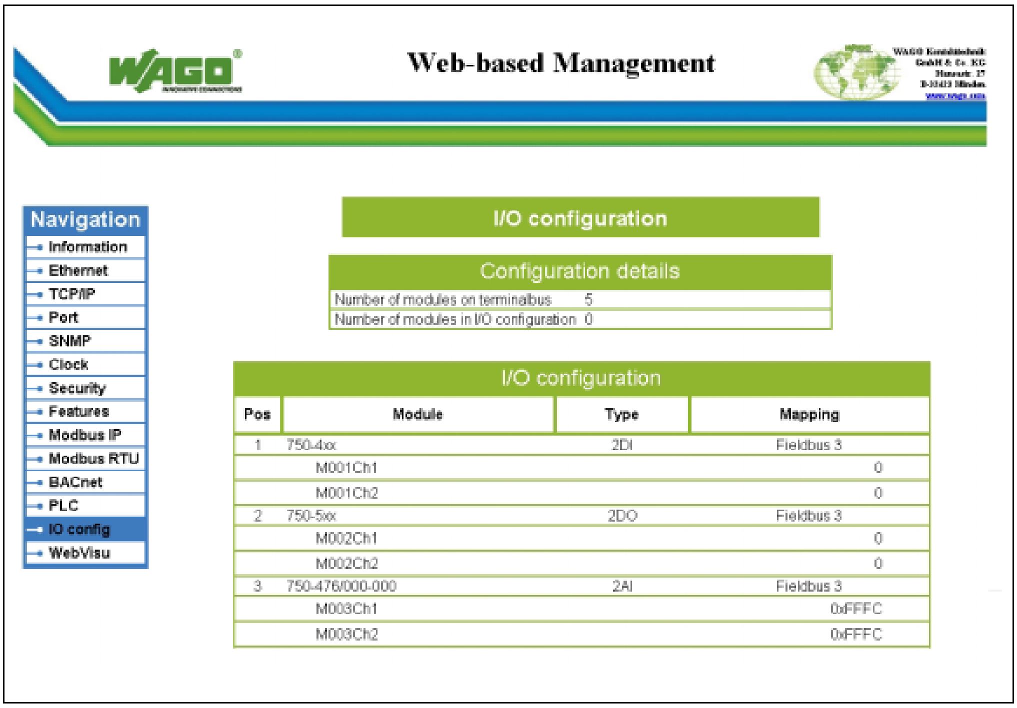


Fig. 3-21: Web-based Management System: I/O config (with process values)

WebVisu

Use the link "WebVisu" to open an HTML page displaying the visualization for your programmed application, provided this has been previously created in WAGO-I/O-PRO CAA and saved to the controller.

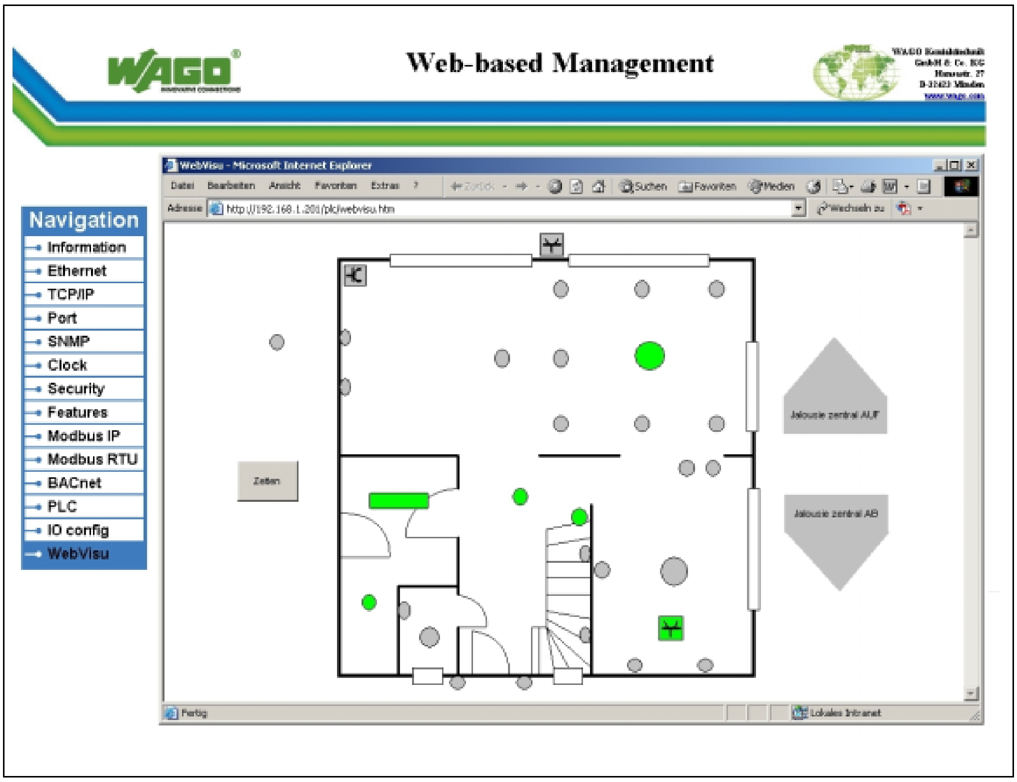


Fig. 3-22: Web-based Management System: WebVisu

A visualization editor is integrated into WAGO-I/O-PRO CAA in order to visualize data of the application programmed with WAGO-I/O-PRO CAA. After selecting the "Web visualization option ("Resources" tab → Target system settings → "Visualization" tab) in WAGO-I/-PRO CAA, an HTML page is created automatically with visualization when compiling the project. A link is then created to this page by the Web-based Management system, so that "WebVisu" is also displayed at this Web page.

You can set "WebVisu" as the starting (default) page. To do this, open the Web page using the "PLC" link and activate the function "Default webpage - Set, webvisu.huml as default." When the Web-based management system is accessed, the "WebVisu" page will be opened instead of the normal "Status Information" page.

Note
The "Webvisu.htm" page does not have any hyperlinks to other Web pages. To deactivate the starting page function, or to access 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.

3.1.9 LED Signaling

For on-site diagnostics, the controller has several LEDs that indicate operational status for both the controller and the entire node.

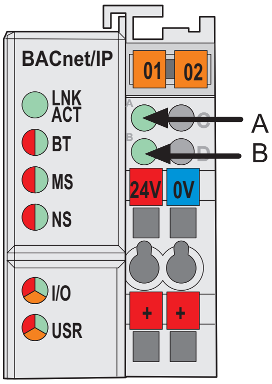


Fig. 3-14:Indicators 750-830 g083002x

Three different groups of LEDs are differentiated.

Fieldbus status (see section 3.1.9.1)
This LED group includes the single and two-color LEDs designated "LINK/ACT (green), "BT" (green), "MS" (red/green) und "NS" (red/green) that indicate the status of fieldbus communication and the operating mode.

Node status (see sections 3.1.9.2 and 3.1.9.3)
The LEDs arranged at the type "I/O" (red/green/orange) and "USR" (red/green/orange) indicate the status of the internal bus and of the fieldbus node. The USR LED can be controlled from an application program in the controller.

Status of power supply (see section 3.1.9.4)
The LEDs A and B signal 24 V power supply (A) and the field side supply, i.e. to the power jumper contacts (B).

3.1.9.1 Fieldbus Status

Communication status via ETHERNET is indicated by the upper-LED group (‘LINK/ACT’, 'BT' ‘MS’, 'NS' and ‘I/O’).

Tab. 3-22: Fieldbus status

LED	Color	Meaning
LNK/ACT	Off	No network connection at Port 1
	Green	Network connection at Port 1 (link)
	Blinking	Data traffic at Port 1 (activity)
BT	Off	No BACnet data traffic
	Green	BACnet data traffic, ready
	Green blinking	Incoming PTP BACnet data packet was accepted as valid and forwarded for further processing
	Red	No BACnet data traffic (Initialization)
MS	Off	No power supply
	Green	The system is operating correctly
	Green flashing	The system is not yet configured
	Red, flashing	The system indicates a recoverable error
	Red	The system indicates an unrecoverable error
	Red/green (orange) flashing	Self test
NS	Off	No IP address has been assigned to the system.
	Green	At least one connection has been set up.
	Green flashing	No connection
	Red, flashing	An error has occurred in the communication over the ETHERNET
	Red	Double IP address in the network
	Red/green (orange) flashing	Self test
I/O	Red/green (orange)	The "I/O" LED indicates the operational status of the node and signals any errors.
USR	Red/green (orange)	The "USR" LED can be selected by a user program in a programmable fieldbus controller.
A	Green	Status of the system power supply
B	Green	Status of power of the power jumper contacts (position of LED determined by production)

3.1.9.2 Node Status – "I/O" LED Blink Code

Tab. 3-23: Node status

LED	Color	Meaning
I/O	Red /green / orange	The "I/O" LED indicates the operational status of the node and signals any errors.

After applying the supply voltage, the controller boots up. The red 'I/O' LED blinks.
After an error-free run-up, the "I/O" LED stays lit as green.

In the event of a failure, the 'I/O' LED will blink continuously.

Detailed error messages are indicated by means of blinking codes. An error is indicated cyclically by up to 3 blinking sequences.

- The error display starts with the first blinking sequence (approx. 10 Hz).
- After a short break, the second blinking sequence starts (approx. 1 Hz). The number of light pulses indicates the **error code**.
- After another break, the third blinking sequence starts (approx. 1 Hz). The number of light pulses indicates the **error argument**.

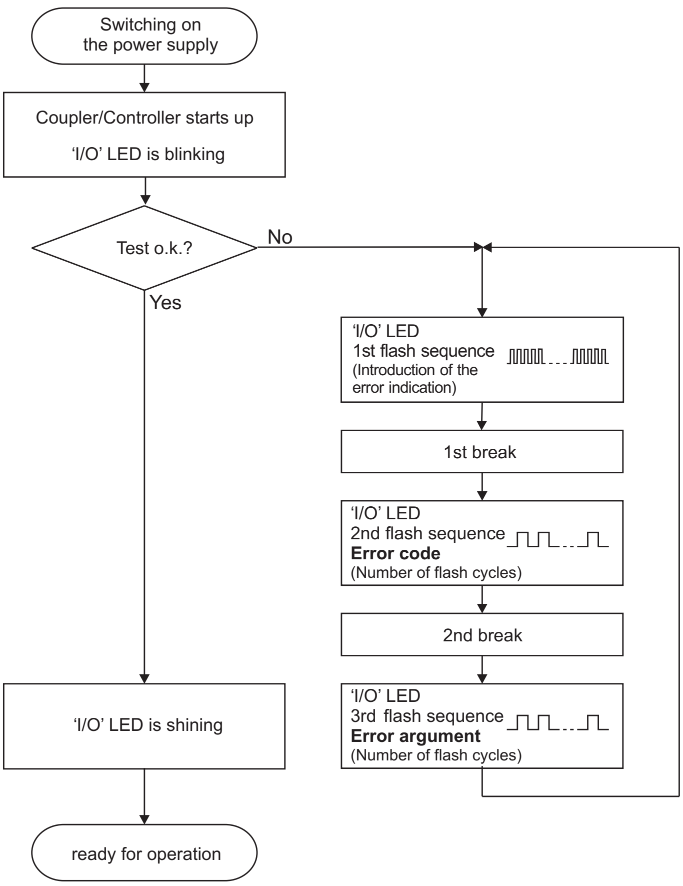


Fig. 3-23: LED signaling of the node status g012111e

After elimination of the error, the controller must be restarted by means of switching the power off and on again.

Tab. 3-24: Signaling of the "I/O" LED

I/O	Meaning
green	Data cycle on the internal bus
off	No data cycle on the internal bus
red	Controller hardware defect
red flashing	During run-up: Internal bus initialized During operation: General internal bus error
red cyclical flashing	Error message for bus module reset and internal error. This error message is evaluated as an error code and error argument on the basis of the blink code.

"I/O" LED Error Messages as Blinking Sequences

Error messages are indicated by three consecutive blinking sequences.

1		2		3
Initiation of error indication	–Pause–	**Error code**	–Pause–	**Error argument**

Tab. 3-25: Error messages as blinking sequences – Error codes 1 through 11

Error code 1: "Hardware and configuration error"		
Error argument	**Error description**	**Solution**
1	Overflow of the internal buffer memory for the inline code.	1. Switch off the power for the node. 2. Reduce the number of bus modules 3. Turn on the power supply again. 4. If the error remains, replace the bus coupler.
2	Bus module(s) with unknown data type	1. Identify the faulty bus module. Turn off the power supply. 2. Plug the end module into the middle of the node and switch the power back on. 3. LED continues to flash? Switch off the power and plug the end module into the middle of the first half of the node (toward the controller). LED not flashing? Switch off the power and plug the end module into the middle of the second half of the node (away from controller). 4. Turn on the power supply again. Repeat this procedure (while halving the step size) until the faulty bus module is detected. 5. Replace the faulty bus module. Inquire about a firmware update for the bus coupler.
3	Invalid check sum in the parameter area of the bus coupler.	1. Switch off the power for the node. 2. Replace the bus coupler and switch the power on again.
4	Fault when writing in the serial EEPROM.	1. Switch off power for the node. 2. Replace the bus coupler and switch the power on again.
5	Fault when reading the serial EEPROM	1. Switch off power for the node. 2. Replace the bus coupler and switch the power on again.
6	Changed bus module configuration found after AUTORESET.	1. Restart the coupler by turning the supply voltage off and on again.

Error code 1: "Hardware and configuration error"		
Error argument	**Error description**	**Solution**
7	Invalid hardware-firmware combination.	1. Switch off power for the node. 2. Replace the bus coupler and switch the power on again.
8	Timeout during serial EEPROM access.	1. Switch off power for the node. 2. Replace the bus coupler and switch the power on again.
9	Bus controller initialization error	1. Switch off power for the node. 2. Replace the bus coupler and switch the power on again.
10	Buffer power failure real-time clock (RTC)	1. Set the clock. 2. Maintain the power supply for the fieldbus coupler for at least 15 minutes in order to charge the Gold cap.
11	Fault during read access to the real-time clock (RTC)	1. Set the clock. 2. Maintain the power supply for the fieldbus coupler for at least 15 minutes in order to charge the Gold cap.
12	Fault during write access to the real-time clock (RTC)	1. Set the clock. 2. Maintain the power supply for the fieldbus coupler for at least 15 minutes in order to charge the Gold cap.
13	Clock interrupt fault	1. Set the clock. 2. Maintain the power supply for the fieldbus coupler for at least 15 minutes in order to charge the Gold cap.
14	Maximum number of gateway or mailbox bus modules exceeded	1. Reduce the number of correspondent modules to a valid number.

Error code 2 -not used-

Error code 3 “Protocol error internal bus”		
Error argument	**Error description**	**Solution**
-	Internal bus communication is faulty, defective component cannot be identified.	1. Are passive power supply modules (750-613) located at the node? First check that these modules are supplied correctly with power. To do so, check the status LEDs.

Error code 3 “Protocol error internal bus”		
Error argument	**Error description**	**Solution**
		2. Are all modules connected correctly or are there any 750-613 bus modules in the node? 3. Switch off the power for the node. 4. Plug the end module in the middle of the node. Turn on the power supply again. 5. LED continues to flash? Switch off the power and plug the end module into the middle of the first half of the node (toward the controller). LED not flashing? Switch off the power and plug the end module into the middle of the second half of the node (away from controller). 6. Turn on the power supply again. 7. Repeat Items 5 and 6 (while halving the step size) until the faulty bus module is detected. 8. Replace the faulty bus module. 9. If there is only one bus module left and the LED continues to flash, either this module or the controller is defective. Replace the faulty component.

Error code 4 "Physical error, internal bus"		
Error argument	**Error description**	**Solution**
-	Internal bus data transmission error or interruption of the internal bus at the bus coupler.	1. Switch off the power for the node. 2. Place a bus module with process data behind the coupler and note the error argument after the power supply is turned on. 3. If no error argument is indicated by the I/O LED, replace the fieldbus coupler. 4. Otherwise identify the faulty bus module. Turn off the power supply. 5. Plug the end module in the middle of the node. Turn on the power supply again.

Error code 4 "Physical error, internal bus"		
Error argument	**Error description**	**Solution**
		6. LED continues to flash? Switch off the power and plug the end module into the middle of the first half of the node (toward the controller). LED not flashing? Switch off the power and plug the end module into the middle of the second half of the node (away from controller). 7. Turn on the power supply again. 8. Repeat procedure 6 and 7 (while halving the step size) until the faulty bus module is detected. 9. Replace the faulty bus module. 10. If there is only one bus module left and the LED continues to flash, then either this module or the controller is defective. Replace the faulty component.
n*	Interruption of the internal bus behind the nth bus module with process data.	1. Turn off the supply voltage of the node, replace the (n+1)th bus module with process data and turn on the supply voltage again.

Error code 5 "Initialization error internal bus"		
Error argument	**Error description**	**Solution**
n*	Error in register communication during internal bus initialization.	1. Switch off the power for the node. 2. Exchange the nth bus module with process data. 3. Turn on the power supply again.

Error code 6 " Node configuration error "		
Error argument	**Error description**	**Solution**
1	Invalid MAC ID	1. Switch off the power for the node. 2. Replace the bus coupler. 3. Turn on the power supply again.
2	Initialization error ETHERNET hardware	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. If the error remains, replace the fieldbus coupler.
3	Initialization error TCP/IP stack	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. If the error remains, replace the fieldbus coupler.
4	Network configuration error (no IP address)	1. Check the settings of the BootP server.

Error code 6 " Node configuration error "		
Error argument	**Error description**	**Solution**
5	Initialization error of an application protocol	1. Restart the fieldbus coupler by turning the power supply off and on again.
6	Maximum process image size exceeded	1. Reduce the number of bus modules
7	IP address of the bus coupler is repeated several times in the network	1. Use an IP address that has not been used in the network
8	Error during process image generation	1. Reduce the number of bus modules on the node
9	The project in WAGO-I/O-PRO CAA differs from the SYM_XML file or the SYM_XML file is missing. The connection between WAGO-I/O-PRO CAA and BACnet is interrupted.	1. Check both the project in WAGO-I/O-PRO CAA and the SYM_XML file. Both must match.

Error code 7 -not used-

Error code 8 -not used-

Error code 9 - not used -

Error code 10 "Error during PLC program processing"		
Error argument	**Error description**	**Solution**
1	Error while applying the PFC runtime system	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. Please contact I/O Support if the error continues.
2	Error while generating the PFC inline code	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. Please contact I/O Support if the error continues.
3	An IEC task has exceeded the maximum runtime, or the call interval for the IEC task could not be maintained (timeout)	1. Check the task configuration with regard to the set call intervals and monitoring times.

Error code 10 "Error during PLC program processing"		
Error argument	**Error description**	**Solution**
4	Error while initializing PFC Web visualization	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. Should the error persist, perform a reset (origin) in WAGO-I/O-PRO, retranslate the project again and reload it to the controller.
5	Error when synchro-nizing the PLC con-figuration with the internal data bus	1. Check the information of the connected modules in the PLC configuration of WAGO-I/O-*PRO* CAA and compare this information with the modules that are actually connected.

Error code 11 "Gateway/Mailbox module error"		
Error argument	**Error description**	**Solution**
1	Too many gateway modules connected	1. Reduce the number of gateway modules
2	Maximum mailbox size exceeded	1. Reduce the size of the mailbox
3	Maximum PA size exceeded due to gate-way modules being connected.	1. Reduce the data width of the gateway modules

* The number of light pulses (n) indicates the position of the bus module. bus modules without data are not counted (e.g. supply modules without diagnostics).

Example: The 13th bus module is removed	
1.	The “I/O” LED starts the error display with the first blinking sequence (approx. 10 Hz).
2.	After the first break, the second blinking sequence starts (approx. 1 Hz). The “I/O” LED blinks four times, indicating error code 4 (data error internal bus).
3.	Afterward, the third blinking sequence will start. The I/O LED blinks twelve times. Error argument 12 means that the internal bus is interrupted behind the twelfth bus module.

3.1.9.3 Node Status – USR-LED

The bottom indicator LED ("USR") is provided for visual output of information about internal bus errors. The activation of the LED from the user program occurs with the functions from the WAGO-I/O-PRO library "Visual.lib".

3.1.9.4 Status supply voltage

Tab. 3-26: Status supply voltage

LED	Color	Meaning
A	green	Status of power – system
B	green	Status of power – power jumper contacts (position of LED determined by production)

The power supply unit of the controller has two green LEDs that indicate the status of the power supply. LED A (left, top) indicates the 24 V supply for the controller.
LED B (left, bottom) and LED C (right, top) indicate the supply of the field side; i.e., the power jumper contacts.

3.1.10 Fault behavior

3.1.10.1 Loss of Fieldbus - MODBUS

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 (see section 4.3.5.1.1), 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.

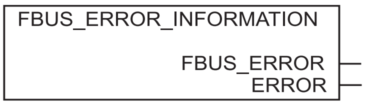


Fig. 3-24: Function block for determining loss of fieldbus, independently of protocol g012926x

'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.

Additional Information
Loss of fieldbus detection through MODBUS protocol:
For detailed information about the watchdog register, refer to section 4.3, "MODBUS Functions", in particular 4.3.5.1.1, "Watchdog (Response on loss of fieldbus)".
Protocol-dependent loss of fieldbus detection:
You can obtain the library 'Mod_com.lib' with the function block 'FBUS_ERROR_INFORMATION' free of cost at the website http://www.wago.com under Downloads → AUTOMATION → WAGO-I/O-PRO Libraries → Mod_com.lib

3.1.10.2 Internal Bus Error

An internal bus failure occurs, for example, if a bus module is removed. If the error occurs during operation, the output modules operate as they do during an internal bus stop.

The "I/O" LED flashes red.

The controller generates an error message (error code and error argument).

If the internal bus failure is resolved, the controller starts up after turning the power off and on again as for a normal startup. The process data is transmitted again and the outputs of the node are set accordingly.

3.1.11 Technical Data

System data	
System data ETHERNET	
Number of controllers	Limited by ETHERNET specification
Transmission medium	Twisted Pair S-UTP 100 W CAT 5
max. length of fieldbus segment	100 m acc. to IEEE 802.3 standard
max. network length	acc. to IEEE 802.3 standard
Baud rate	10/100 Mbit/s
Fieldbus coupler connection	1 x RJ45, 1 x RS232
Protocols	BACnet/IP, BACnet PTP, MODBUS/TCP (UDP), HTTP, BootP, DHCP, DNS, SNTP, FTP, SNMP V1, SMTP
Serial system data (BACnet PTP)*	* available from software version 2
Baud rate	9600 baud ... 115 200 baud
max. length of fieldbus segment	15 m depending on the baud rate / on the cable (at 19200 baud)
Fieldbus coupler connection	1 x D-Sub 9 socket
Programming	WAGO-I/O-*PRO* CAA
IEC 61131-3-3	IL, LD, FBD (CFC), ST, FC
BACnet device profile	B-BC (BACnet-Building-Controller)
Standards and regulations (see chapter 2.2)	
EMC CE Immunity to interference	in acc. with EN 61000-6-2 (2005)
EMC CE Emission of interference	in acc. with EN 61000-6-3 (2001) +A11:2004
Vibration resistance	acc. to IEC 60068-2-6
Shock resistance	acc. to IEC 60068-2-27
Degree of protection	IP 20
Approvals (see chapter 2.2)	
cULus	$_{C}UL_{US}$ (UL508)
CE	Conformity marking
	BACnet conformity test in preparation
Accessories	
PC software	WAGO BACnet Configurator
Miniature WSB Quick marking system	
Technical Data	
Number of bus modules with bus extension	64 250
Configuration	via PC
Program memory	512 Kbytes
Data memory	256 Kbytes
Non-volatile memory (retain)	24 Kbytes (16 Kbytes retain, 8 Kbytes flag)
Voltage supply	DC 24 V (-25 % ... +30 %)
Input current $_{max}$	500 mA at 24 V
Efficiency of the power supply	87 %
Internal current consumption	300 mA at 5 V
Total current for bus modules	1700 mA at 5 V
Isolation	500 V system/supply

Technical Data	
Voltage via power jumper contacts	DC 24 V (-25 % ... + 30 %)
Current via power jumper contacts $_{max}$	DC 10 A
BACnet implementation acc. to	DIN EN ISO 16484-5 =ANSI/ASHRAE 135-2004
Fieldbus (MODBUS/TCP)	
Input process image $_{max}$	2 Kbytes
Output process image $_{max}$	2 Kbytes
Input variables $_{max}$	512 bytes
Output variables $_{max}$	512 bytes
Operating temperature	0 °C ... +55 °C
Wire connection	CAGE CLAMP®
Cross sections	0.08 mm² ... 2.5 mm² / AWG 28 ... 14
Stripped lengths	8 ... 9 mm / 0.33 in
Dimensions (mm) W x H x L	51 x 65* x 100 (*from upper edge of mounting rail)
Weight	approx. 188 g
Storage temperature	-25 °C ... +85 °C
Relative humidity (without condensation)	95 %

3.1.12 BACnet Building Controller (B-BC)

The BACnet Standard 135-2004 describes six 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.

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 **4.2**.

The WAGO BACnet/IP Controller 750-830 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 and 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

Tab. 3-27: BIBBs of the B-BC shows the minimum requirement for the BIBBs for the B-BC in general as well as additional BIBBs implemented by the WAGO BACnet/IP Controller.

Tab. 3-27: BIBBs of the B-BC

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

3.1.12.1.1.1 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 COVP-A (DS-COVP-A)

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

BACnet Service	Requests	Execute
SubscribeCOVProperty	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 COVP-B (DS-COVP-B)

Device B is a provider of COV data on any property of a specific object for Device A.

BACnet Service	Requests	Execute
SubscribeCOVProperty		x
ConfirmedCOVNotification	x	
UnconfirmedCOVNotification	x	

Devices that comply with DS-COVP-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-Unsolicited-A (DS-COVU-A)

The A device processes unsolicited COV data from device B.

BACnet Service	Requests	Execute
UnconfirmedCOVNotification		x

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

The B device generates unsolicited COV notifications.

BACnet Service	Requests	Execute
UnconfirmedCOVNotification	x	

3.1.12.1.1.2 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-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-Enrollment Summary-B (AE-ESUM-B)

Device B provides event enrollments to device A.

BACnet Service	Requests	Execute
GetEnrollmentSummary		x

3.1.12.1.1.3 Scheduling BIBBs

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

Scheduling A (SCHED-A)

Device A processes schedules and the calendar from device B. Device A must support the BIBBs DS-RP-A and DS-WP-A.

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.

3.1.12.1.1.4 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.

3.1.12.1.1.5 Device and 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-A (DM-DOB-A)

The A device seeks address information about BACnet objects.

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

Network Management-Connection Establishment-A (NM-CE-A)

The A device commands a half-router to establish and terminate connections as needed for communication with other devices.

BACnet Network Layer Message	Requests	Execute
Establish-Connection-To-Network	x	
Disconnect-Connection-To-Network	x	

3.1.12.2 "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
A maximum of 1000 objects can be created in the 750-830 BACnet/IP Controller.

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

Additional Information
You can program objects that are not automatically created in the WAGO-I/O-*PRO* CAA. You can obtain additional information on the Internet under Documentation → 750-830 → Additional Information → BACnet - Objects, Properties, Services

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.

3.1.12.2.1 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 Tab. 3-28. The properties are described in section **6**.

Tab. 3-28: 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	-

*corresponds to the item number of the module

3.1.12.2.2 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 Tab. 3-29. The properties are described in section **6**.

Tab. 3-29: 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 BACnet-TimeStamp	UNSPECIFIED	-

*corresponds to the item number of the module

3.1.12.2.3 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 Tab. 3-30. The properties are described in section **6**.

Tab. 3-30: 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	-

Property	Data type:	Default Value	Writable via BACnet by means of
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	-

*corresponds to the item number of the module

3.1.12.2.4 Binary Output Object

The Binary Output object defines a standardized object whose properties represent the externally visible characteristics of a binary output. A "binary output" 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 Tab. 3-31. The properties are described in section **6**.

Tab. 3-31: 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	-
Elapsed_Active_Time	Unsigned32	0	-
Time_Of_Active_Time_Reset	BACnetDateTime	UNSPECIFIED	-
Minimum_Off_Time	Unsigned32	1	WriteProperty

Property	Data type:	Default Value	Writable via BACnet by means of
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	BACnetEventTransitionBits	'111'	WriteProperty
Acked_Transitions	BACnetEventTransitionBits	'111'	-
Notify_Type	BACnetNotifyType	Alarm (0)	WriteProperty
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	UNSPECIFIED	-

*corresponds to the item number of the module

3.1.12.2.5 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 Tab. 3-32. The properties are described in section **6**.

Tab. 3-32: 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 for BACnetCalendarEntry	-	WriteProperty

3.1.12.2.6 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 Tab. 3-33. The properties are described in section **6**.

Tab. 3-33: 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 Kontakttechnik GmbH & Co. KG"	-
Vendor_Identifier	Unsigned16	222	-
Model_Name	CharacterString	750-830	-
Firmware_Revision	CharacterString	-	-
Application_Software_Version	CharacterString	-	-
Location	CharacterString	""	WriteProperty
Description	CharacterString	""	WriteProperty
Protocol_Version	Unsigned	-	-
Protocol_Revision	Unsigned	-	-
Protocol_Services_Supported	BACnetServicesSupported	-	-
Protocol_Object_Types_Supported	BACnetObjectTypesSupported	-	-
Object_List	BACnetARRAY[N] of the BACnetObjectIdentifier	-	-
Max_APDU_Length_Accepted	Unsigned	1476	-
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	-

Property	Data type:	Default Value	Writable via BACnet by means of
Device_Address_Binding	List for BACnetAddress-Binding	-	-
Database_Revision	Unsigned	-	-
Configuration_Files	BACnetARRAY[N] of the BACnetObjectIdentifier	-	-
Last_Restore_Time	BACnetTimeStamp	UNSPECIFIED	-
Backup_Failure_Timeout	Unsigned16	10	-
Active_COV_Subscriptions	List for BACnetCOVSub-scription	-	-

3.1.12.2.7 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 Tab. **3-34**. The properties are described in section **6**.

Tab. 3-34: 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	-
Read_Only	BOOLEAN	TRUE	WriteProperty
File_Access_Method	BACnetFileAccessMethod	STREAM_ ACCESS	-

3.1.12.2.8 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 Tab. **3-35**. The properties are described in section **6**.

Tab. 3-35: Properties of the Schedule Object

Properties	Data type:	Based on IEC Data Types	Default Value	Writable via BACnet by means of
Object_Identifier	BACnetObjectIdentifier	BACnetObjectIdentifier	-	-
Object_Name	CharacterString	STRING	"ScheduleX"	-
Object_Type	BACnetObjectType	BACnetObjectType	SCHEDULE	-
Present_Value	Any	ARRAY[0..10] OF BYTE	-	-
Description	CharacterString	STRING	""	WriteProperty
Effective_Period	BACnetDateRange	BACnetDateRange	UNSPECIFIED	WriteProperty
Weekly_Schedule	BACnetARRAY[7]of BACnetDailySchedule	ARRAY [1..7] OF BACnetDailySchedule	-	WriteProperty
Exception_Schedule	BACnetARRAY[N]of BACnetSpecialEvent	ARRAY [1..7] OF BACnetSpecialEvent	-	WriteProperty
Schedule_Default	Any	ARRAY[0..10] OF BYTE	-	WriteProperty
List_Of_Object_ Property_References	List of BACnetDevice ObjectPropertyReference	ARRAY [0..3] OF BACnetDeviceObject-PropertyReference;	-	WriteProperty

Properties	Data type:	Based on IEC Data Types	Default Value	Writable via BACnet by means of
Priority_For_Writing	Unsigned(1..16)	BYTE	16	WriteProperty
Status_Flags	BACnetStatusFlags	BACnetStatusFlags	‘0000’	-
Reliability	BACnetReliability	BACnetReliability	-	-
Out_Of_Service	BOOLEAN	BOOL	-	WriteProperty

4 Fieldbus Communication

4.1 ETHERNET

4.1.1 General

ETHERNET is a technology, which has been proven and established as an effective means of data transmission in the field of information technology and office communication. Within a short time ETHERNET has also made a successful breakthrough in the area of private PC networks throughout the world.

This technology was developed in 1972 by Dr. Robert M. Metcalfe, David R. Boggs, Charles Thacker, Butler W. Lampson, and Xerox (Stanford, Ct.). Standardization (IEEE 802.3) took place in 1983.

ETHERNET predominantly uses coaxial cables or twisted pair cables as a transmission medium. Connection to ETHERNET, often already existing in networks, (LAN, Internet) is easy and the data exchange at a transmission rate of 10 Mbps or for some couplers/controllers also 100 Mbps is very fast.

ETHERNET has been equipped with higher level communication software in addition to standard IEEE 802.3, such as TCP/IP (Transmission Control Protocol / Internet Protocol) to allow communication between different systems. The TCP/IP protocol stack offers a high degree of reliability for the transmission of information.

In the ETHERNET based (programmable) fieldbus couplers and controllers developed by WAGO, usually various application protocols have been implemented on the basis of the TCP/IP stack.

These protocols allow the user to create applications (master applications) with standardized interfaces and transmit process data via an ETHERNET interface.

In addition to a series of management and diagnostic protocols, fieldbus specific application protocols are implemented for control of the module data, depending upon the coupler or controller, e. g. MODBUS TCP (UDP), EtherNet/IP, BACnet/IP, KNXNET/IP, PROFINET, Powerlink, SERCOS III or others.

Information such as the fieldbus node architecture, network statistics and diagnostic information is stored in the ETHERNET (programmable) fieldbus couplers and controllers and can be viewed as HTML pages via a web browser (e.g., Microsoft Internet-Explorer, Netscape Navigator) being served from the HTTP server in the couplers and controllers.

Furthermore, depending on the requirements of the respective industrial application, various settings such as selection of protocols, TCP/IP, internal clock and security configurations can be performed via the web-based management system. However, you can also load web pages you have created yourself into the couplers/controllers, which have an internal file system, using FTP.

The WAGO ETHERNET TCP/IP fieldbus node does not require any additional master components other than a PC with a network card. So, the fieldbus node can be easily connected to local or global networks using the fieldbus connection. Other networking components such as hubs, switches or repeaters can also be used. However, to establish the greatest amount of "determinism" a switch is recommended.

The use of ETHERNET as a fieldbus allows continuous data transmission between the plant floor and the office. Connection of the ETHERNET TCP/IP fieldbus node to the Internet even enables industrial processing data for all types of applications to be called up world-wide. This makes site independent monitoring, visualization, remote maintenance and control of processes possible.

4.1.2 Network Architecture – Principles and Regulations

A simple ETHERNET network is designed on the basis of one PC with a network interface card (NI), one crossover connection cable (if necessary), one ETHERNET fieldbus node and one 24 V DC power supply for the coupler/controller voltage source.

Each fieldbus node consists of a (programmable) fieldbus coupler or controller and a number of needed I/O modules.

Sensors and actuators are connected to the digital or analog I/O modules on the field side. These are used for process signal acquisition or signal output to the process, respectively.

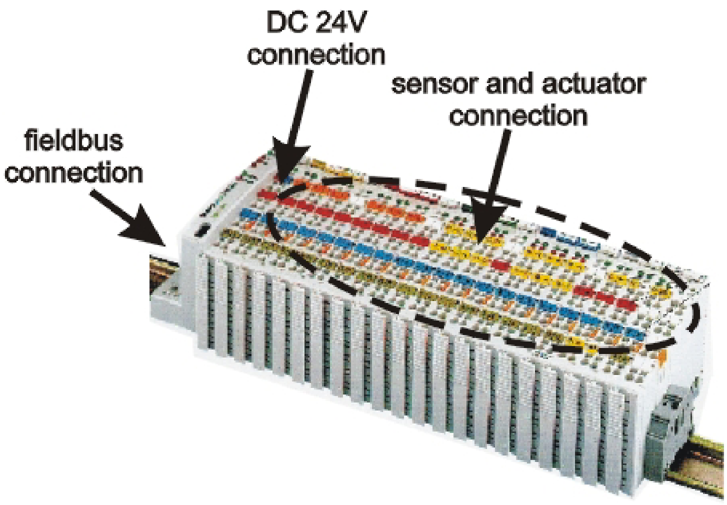


Fig. 4-1. Connection Example and Principle of a Fieldbus Node for a Network Architecture
1Netwerkknotene

Fieldbus communication between master application and (programmable) fieldbus coupler or controller takes place using the implemented fieldbus specific application protocol, e. g. MODBUS TCP (UDP), EtherNet/IP, BACnet, KNXNET/IP, PROFINET, Powerlink, SERCOS III or others.

4.1.2.1 Transmission Media

General ETHERNET transmission standards

For transmitting data the ETHERNET standard supports numerous technologies with various parameters (e.g., transmission speed, medium, segment length and type of transmission).

Tab. 4-1: ETHERNET Transmission Standards

1Base5	Uses a 24 AWG UTP (twisted pair cable) for a 1Mbps baseband signal for distances up to 500 m (250 m per segment) in a physical star topology.
10Base2	Uses a 5 mm 50 Ohm coaxial cable for a 10Mbps baseband signal for distances of up to 185 m in a physical bus topology (often referred to as Thin ETHERNET or ThinNet).
10Base5	Uses a 10 mm 50 Ohm coaxial cable for a 10Mbps baseband signal for distances of up to 500 m in a physical bus topology (often referred to as Thick ETHERNET).
10BaseF	Uses a fiber-optic cable for a 10Mbps baseband signal for distances of up to 4 km in a physical star topology. (There are three sub-specifications: 10BaseFL for fiber-optic link, 10BaseFB for fiber-optic backbone and 10BaseFP for fiber-optic passive).
10BaseT	Uses a 24 AWG UTP or STP/UTP (twisted pair cable) for a 10Mbps baseband signal for distances up to 100 m in a physical star topology.
10Broad36	Uses a 75 Ohm coaxial cable for a 10Mbps baseband signal for distances of up to 1800 m (or 3600 m with double cables) in a physical bus topology.
100BaseTX	Specifies a 100 Mbps transmission with a twisted pair cable of Category 5 and RJ45-connectors. A maximum segment of 100 meters may be used.

Beyond that there are still further transmission standards, for example: 100BaseT4 (Fast ETHERNET over twisted conductors), 100BaseFX (Fast ETHERNET over fiber-optic cables) or P802.11 (Wireless LAN) for a wireless transmission.

The media types are shown with their IEEE shorthand identifiers. The IEEE identifiers include three pieces of information.
The first item, for example, “10”, stands for the media.
The third part of the identifier provides a rough indication of segment type or length. For thick coaxial cable, the “5” indicates a 500 meter maximum length allowed for individual thick coaxial segments. For thin coaxial cable, the “2” is rounded up from the 185 meter maximum length for individual thin coaxial segments. The “T” and “F” stand for ‘twisted pair’ and ‘fiber optic’, and simply indicate the cable type.

10BaseT, 100BaseTX

Either the 10BaseT standard or 100BaseTX can be used for the WAGO ETHERNET fieldbus node.
The network architecture is very easy and inexpensive to assemble with S-UTP cable as transmission medium or with cables of STP type.
Both types of cable can be obtained from any computer dealer.

S-UTP cable (screened unshielded twisted pair) is single-shielded cable of Category 5 with overall shield surrounding all twisted unshielded conductor pairs and an impedance of 100 ohm.
STP cable (shielded twisted pair) is cable of Category 5 with stranded and individually shielded conductor pairs; no overall shield is provided.

Wiring of the fieldbus nodes

Maybe, a crossover cable is required for direct connection of a fieldbus node to the network card of the PC.

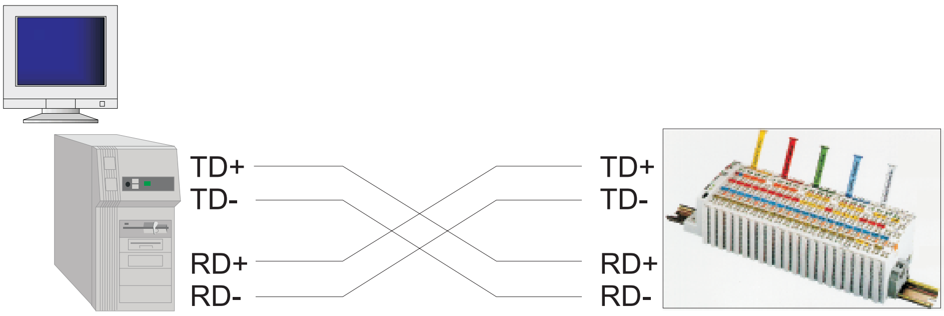


Fig. 4-2: Direct Connection of a Node with Crossover Cable g012906d

If several fieldbus nodes are to be connected to a network card, the fieldbus nodes can be connected via an ETHERNET switch or hub with straight through/parallel cables.

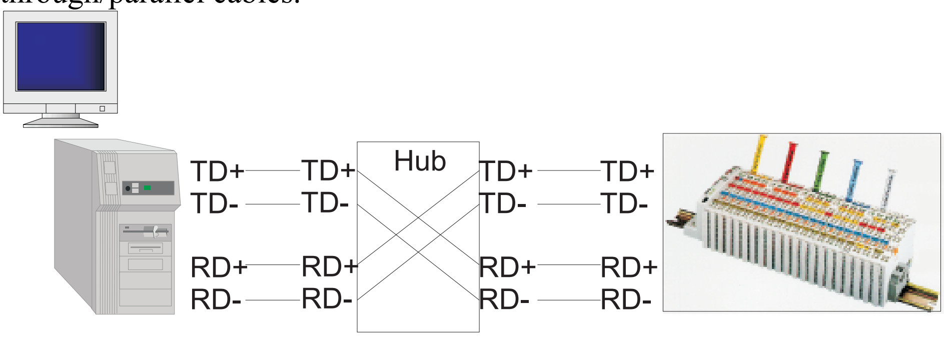


Fig. 4-3: Connection of a Node by means of a Hub with Parallel cables g012908d

An ETHERNET switch is a device that allows all connected devices to transmit and receive data with each other. The switch can also be viewed as a “data traffic cop” where the hub “polices” the data coming in and going out of the individual ports, so the data will only be transmitted to the required node. WAGO recommends using a switch rather then a hub, this will allow for a more deterministic architecture.

Attention
The cable length between the node and the hub cannot be longer than 100 m (328 ft.) without adding signal conditioning systems (i.e., repeaters). Various possibilities are described in the ETHERNET standard for networks covering larger distances.

4.1.2.2 Network Topologies

In the case of 10BaseT, or 100BaseTX several stations (nodes) are connected using a star topology according to the 10BaseT ETHERNET Standard.

Therefore, this manual only deals with the star topology, and the tree topology for larger networks in more detail.

Star Topology

A star topology consists of a network in which all nodes are connected to a central point via individual cables.

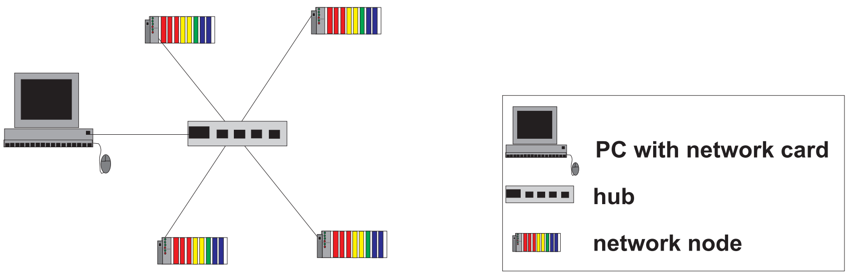


Fig. 4-4: Star Topology G012903e

A star topology offers the advantage of allowing the extension of an existing network. Stations can be added or removed without network interruption. Moreover, in the event of a defective cable, only the network segment and the node connected to this segment is impaired. This considerably increases the fail-safe of the entire network.

Tree Topology

The tree topology combines characteristics of linear bus and star topologies. It consists of groups of star-configured workstations connected to a linear bus backbone cable. Tree topologies allow for the expansion of an existing network, and enables schools, etc. to configure a network to meet their needs.

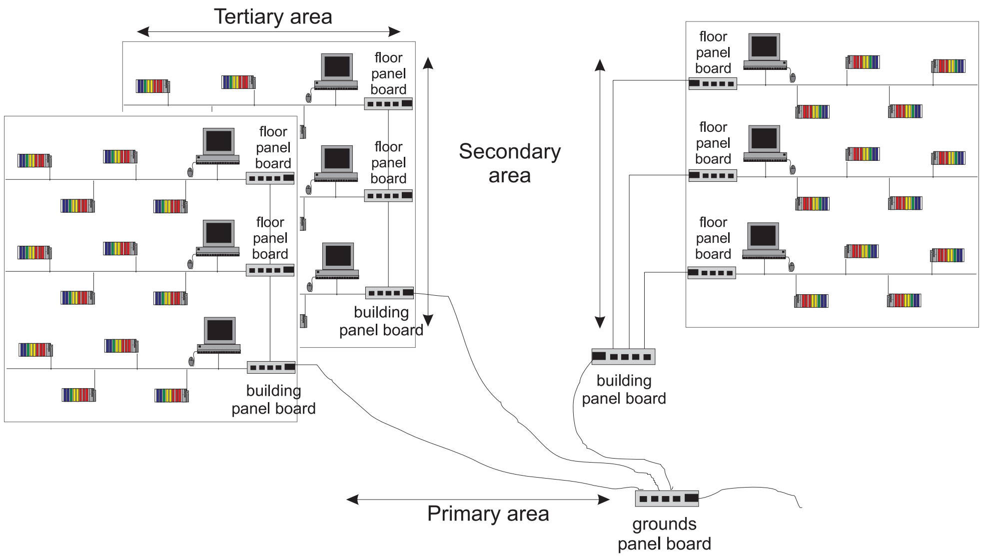


Fig. 4-5: Tree Topology G012904e

5-4-3 Rule

A consideration in setting up a tree topology using ETHERNET protocol is the
5-4-3 rule. One aspect of the ETHERNET protocol requires that a signal sent out on the network cable must reach every part of the network within a specified length of time. Each concentrator or repeater that a signal goes through adds a small amount of time. This leads to the rule that between any two nodes on the network there can only be a maximum of 5 segments connected through 4 repeators/concentrators. In addition, only 3 of the segments may be populated (trunk) segments if they are made of coaxial cable. A populated segment is one that has one or more nodes attached to it. In Figure 5-5, the 5-4-3 rule is adhered to. The furthest two nodes on the network have 4 segments and 3 repeators/concentrators between them.

This rule does not apply to other network protocols or ETHERNET networks where all fiber optic cabling or a combination of a backbone with UTP cabling is used. If there is a combination of fiber optic backbone and UTP cabling, the rule is simply translated to 7-6-5 rule.

Cabling guidelines

"Structured Cabling" specifies general guidelines for network architecture of a LAN, establishing maximum cable lengths for the grounds area, building and floor cabling.

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

The cabling standards define a domain covering a geographical area of 3 km and for an office area of up to 1 million square meters with 50 to 50,000 terminals. In addition, they describe recommendations for setting up of a cabling system.

Specifications may vary depending on the selected topology, the transmission media and coupler modules used in industrial environments, as well as the use of components from different manufacturers in a network. Therefore, the specifications given here are only intended as recommendations.

4.1.2.3 Coupler Modules

There are a number of hardware modules that allow for flexible arrangement for setting up an ETHERNET network. They also offer important functions, some of which are very similar.

The following table defines and compares these modules and is intended to simplify the correct selection and appropriate application of them.

Module	Characteristics/application	ISO/OSI layer
Repeater	Amplifier for signal regeneration, connection on a physical level.	1
Bridge	Segmentation of networks to increase the length.	2
Switch	Multiport bridge, meaning each port has a separate bridge function. Logically separates network segments, thereby reducing network traffic. Consistent use makes ETHERNET collision-free.	2 (3)
Hub	Used to create star topologies, supports various transmission media, does not prevent any network collisions.	2
Router	Links two or more data networks. Matches topology changes and incompatible packet sizes (e.g. used in industrial and office areas).	3
Gateway	Links two manufacturer-specific networks which use different software and hardware (i.e., ETHERNET and Interbus-Loop).	4-7

Tab. 4-2: Comparison of Coupler Modules for Networks

4.1.2.4 Transmission Mode

Some ETHERNET based WAGO couplers/controllers support both 10Mbit/s and 100Mbit/s for either full or half duplex operation. To guarantee a safe and fast transmission, both these couplers/controllers and their link partners must be configured for the same transmission mode.

Attention
A faulty configuration of the transmission mode may result in a link loss condition, a poor network performance or a faulty behavior of the coupler/controller.

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

- Static configuration
- Dynamic configuration

4.1.2.4.1 Static Configuration of the Transmission Mode

Using static configuration, both link partners are set to static transmission rate and duplex mode. The following configurations are possible:

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

4.1.2.4.2 Dynamic Configuration of the Transmission Mode

The second configuration option is the autonegotiation mode which is defined in the IEEE 802.3u standard. Using this mode, the transmission rate and the duplex mode are negotiated dynamically between both communication partners. Autonegotiation allows the device to automatically select the optimum transmission mode.

Attention
To ensure a correct dynamic configuration process, the operation mode for the autonegotiation of both communication partners must be supported and activated.

4.1.2.4.3 Errors Occurring when Configuring the Transmission Mode

Invalid configurations are listed below:

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

4.1.2.5 Important Terms

Data security

If an internal network (Intranet) is to be connected to the public network (e.g., the Internet) then data security is an extremely important aspect.

Undesired access can be prevented by a **Firewall**.
Firewalls can be implemented in software or network components. They are interconnected in a similar way to routers as a switching element between Intranets and the public network. Firewalls are able to limit or completely block all access to the other networks, depending on the access direction, the service used and the authenticity of the network user.

Real-time ability

Transmission above the fieldbus system level generally involves relatively large data quantities. The permissible delay times may also be relatively long (0.1...10 seconds).
However, real-time behavior within the fieldbus system level is required for ETHERNET in industry.
In ETHERNET it is possible to meet the real-time requirements by restricting the bus traffic (< 10 %), by using a master-slave principle, or also by implementing a switch instead of a hub.
MODBUS/TCP is a master/slave protocol in which the slaves only respond to commands from the master. When only one master is used, data traffic over the network can be controlled and collisions avoided.

Shared ETHERNET

Several nodes linked via a hub share a common medium. When a message is sent from a station, it is broadcast throughout the entire network and is sent to each connected node. Only the node with the correct target address processes the message. Collisions may occur and messages have to be repeatedly transmitted as a result of the large amount of data traffic. The delay time in a Shared ETHERNET cannot be easily calculated or predicted.

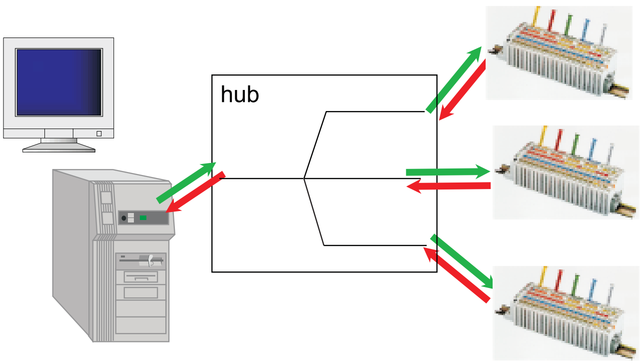


Fig. 4-6: Principle of Shared ETHERNET G012910e

Deterministic ETHERNET

The TCP/IP software or the user program in each subscriber can limit transmittable messages to make it possible to determine real-time requirements. At the same time the maximum medium message rate (datagrams per second), the maximum medium duration of a message, and the minimum time interval between the messages (waiting time of the subscriber) is limited.

Therefore, the delay time of a message is predictable.

Switched ETHERNET

In the case of Switched ETHERNET, several fieldbus nodes are connected by a switch. When data from a network segment reaches the switch, it saves the data and checks for the segment and the node to which this data is to be sent. The message is then only sent to the node with the correct target address. This reduces the data traffic over the network, extends the bandwidth and prevents collisions. The runtimes can be defined and calculated, making the Switched ETHERNET deterministic.

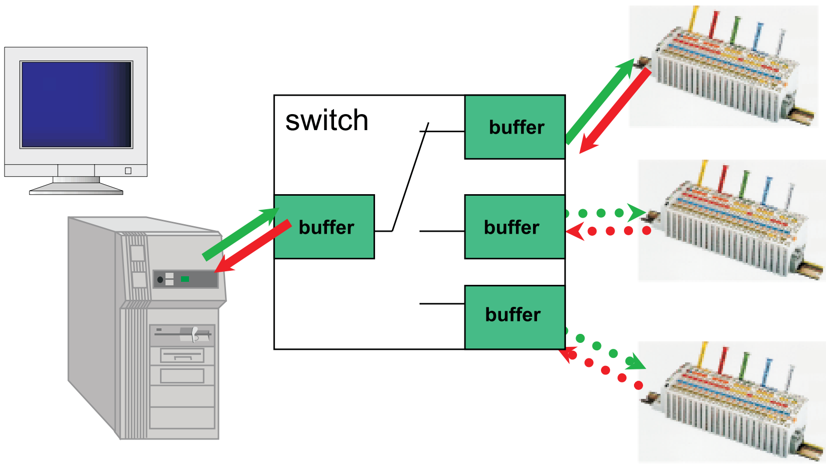


Fig. 4-7: Principle of Switched ETHERNET G012909e

4.1.3 Network Communication

Fieldbus communication between master application and (programmable) fieldbus coupler or controller usually takes place using an implemented field-bus specific application protocol, e. g. MODBUS TCP (UDP), EtherNet/IP, BACnet, KNXNET/IP, PROFINET, Powerlink, SERCOS III or others.

The protocol layer model helps with an example (MODBUS and EtherNet/IP) to explain the classification and interrelationships between the communication and application protocols.
In this example, the fieldbus communication can take place using either the MODBUS protocol or EtherNet/IP.

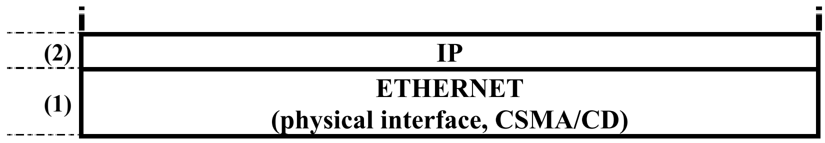
4.1.3.1 Protocol layer model

(1) ETHERNET:

The ETHERNET hardware forms the basis for the physical exchange of data. The exchanged data signals and the bus access procedure CSMA/CD are defined in a standard.

(2) IP:

For the communication the Internet Protocol (IP) is positioned above the ETHERNET hardware. This bundles the data to be transmitted in packets along with sender and receiver address and passes these packets down to the ETHERNET layer for physical transmission. At the receiver end, IP accepts the packets from the ETHERNET layer and unpacks them.

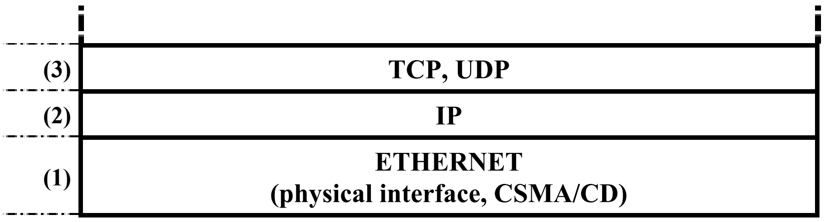


(3) TCP, UDP:

a) TCP: (Transmission Control Protocol)
The TCP protocol, which is positioned above the IP layer, monitors the transport of the data packets, sorts their sequence and sends repeat requests for missing packets. TCP is a connection-oriented transport protocol.
The TCP and IP protocol layers are also jointly described as the TCP/IP protocol stack or TCP/IP stack.

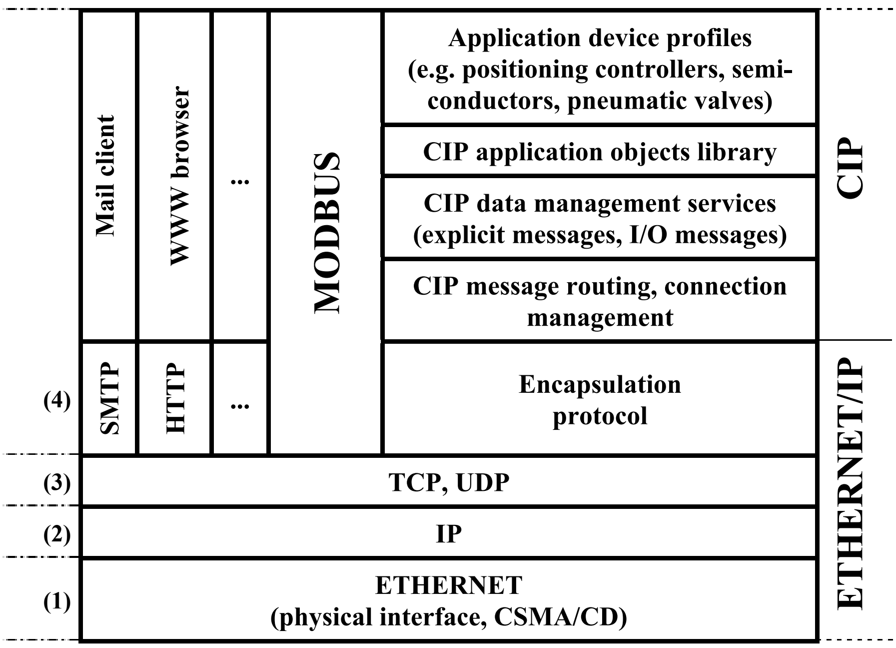
b) UDP: (User Datagram Protocol)
The UDP layer is also a transport protocol like TCP, and is arranged above the IP layer. In contrast to the TCP protocol, UDP is not connection oriented. That means there are no monitoring mechanisms for data exchange between sender and receiver.
The advantage of this protocol is in the efficiency of the transmitted

data and the resultant increase in processing speed.
Many programs use both protocols. Important status information is sent via the reliable TCP connection, while the main stream of data is sent via UDP.



(4) Management, Diagnostic and Application Protocols:

Positioned above the TCP/IP stack or UDP/IP layer are correspondingly implemented management, diagnostic and application protocols that provide services that are appropriate for the application. For the management and diagnostic, these are, for example, SMTP (Simple Mail Transport Protocol) for e-mails, HTTP (Hypertext Transport Protocol) for www browsers and some others.
In this example, the protocols MODBUS/TCP (UDP) and EtherNet/IP are implemented for use in industrial data communication.
Here the MODBUS protocol is also positioned directly above TCP (UDP)/IP; EtherNet/IP, on the other hand, basically consists of the protocol layers ETHERNET, TCP and IP with an encapsulation protocol positioned above it. This serves as interface to CIP (Control and Information Protocol).
DeviceNet uses CIP in the same way as EtherNet/IP. Applications with DeviceNet device profiles can therefore be very simply transferred to EtherNet/IP.



4.1.3.2 Communication Protocols

In addition to the ETHERNET standard, the following important communication protocols are implemented in the WAGO ETHERNET based (programmable) fieldbus couplers and controllers:

- IP Version 4 (Raw-IP and IP-Multicast)
- TCP
- UDP
- ARP

The following diagram is intended to explain the data structure of these protocols and how the data packets of the communication protocols ETHERNET, TCP and IP with the adapted application protocol MODBUS nested in each other for transmission. A detailed description of the tasks and addressing schemes of these protocols is contained in the following.

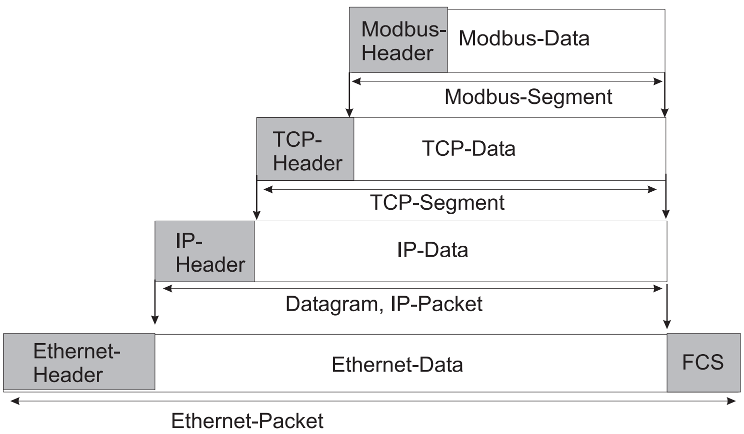


Fig. 4-8: Communication Protocols G012907e

4.1.3.2.1 ETHERNET

ETHERNET address (MAC-ID)

Each WAGO ETHERNET (programmable) fieldbus coupler or controller is provided from the factory with a unique and internationally unambiguous physical ETHERNET address, also referred to as MAC-ID (Media Access Control Identity). This can be used by the network operating system for addressing on a hardware level.
The address has a fixed length of 6 Bytes (48 Bit) and contains the address type, the manufacturer's ID, and the serial number.
Examples for the MAC-ID of a WAGO ETHERNET fieldbus coupler (hexadecimal): 00_H-30_H-DE_H-00_H-00_H-01_H.
ETHERNET does not allow addressing of different networks.
If an ETHERNET network is to be connected to other networks, higher-ranking protocols have to be used.

Attention
If you wish to connect one or more data networks, routers have to be used.

ETHERNET Packet

The datagrams exchanged on the transmission medium are called "ETHERNET packets" or just "packets". Transmission is connectionless; i.e. the sender does not receive any feedback from the receiver. The data used is packed in an address information frame. The following figure shows the structure of such a packet.

Preamble	ETHERNET Header	ETHERNET Data	Check sum
8 Byte	14 Byte	46-1500 Byte	4 Byte

Fig. 4-9: ETHERNET-Packet

The preamble serves as a synchronization between the transmitting station and the receiving station. The ETHERNET header contains the MAC addresses of the transmitter and the receiver, and a type field.
The type field is used to identify the following protocol by way of unambiguous coding (e.g., 0800_{hex} = Internet Protocol).

4.1.3.2.1.1 Channel access method

In the ETHERNET Standard, the fieldbus node accesses the bus using CSMA/CD (Carrier Sense Multiple Access/ Collision Detection).

Carrier Sense:	The transmitter senses the bus.
Multiple Access:	Several transmitters can access the bus.
Collision Detection:	A collision is detected.

Each station can send a message once it has established that the transmission medium is free. If collisions of data packets occur due to several stations transmitting simultaneously, CSMA/CD ensures that these are detected and the data transmission is repeated.

However, this does not make data transmission reliable enough for industrial requirements. To ensure that communication and data transmission via ETHERNET is reliable, various communication protocols are required.

4.1.3.2.2 IP-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 addresses

To allow communication over the network each fieldbus node requires a 32 bit Internet address (IP address).

Attention
Internet addresses have to be unique throughout the entire interconnected networks.

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: Byte1, Host-ID: Byte2 - Byte4)

e.g.: 101 . 16 . 232 . 22

01100101		00010000	11101000	00010110
0	Net-ID	Host-ID		

The highest bit in Class A networks is always ‘0’.
Meaning 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: Byte1 - Byte2, Host-ID: Byte3 - Byte4)

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

10110101		00010000	11101000	00010110
10	Net-ID		Host-ID	

The highest bits in Class B networks are always ’10’.
Meaning 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: Byte1 - Byte3, Host-ID: Byte4)

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’.
Meaning 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

	Address range of the subnetwork	Possible number of networks	Subscribers per network
Class A	1.XXX.XXX.XXX - 126.XXX.XXX.XXX	127 (2^7)	Ca. 16 Million (2^{24})
Class B	128.000.XXX.XXX - 191.255.XXX.XXX	Ca. 16 thousand (2^{14})	Ca 65 thousand (2^{16})
Class C	192.000.000.XXX - 223.255.255.XXX	Ca. 2 million (2^{21})	254 (2^8)

Each WAGO ETHERNET (programmable) 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.

Attention
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 Inter*NIC* (International Network Information Center).

Attention
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.

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

Fig. 4-10: Class B address with Field for Subnet 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:

255	.0	.0	.0

Class B Subnet mask:

255	.255	.0	.0

Class C Subnet mask:

255	.255	.255	.0

Depending on the subnet division the subnet masks may, however, contain other values beyond 0 and 255, such as 255.255.255.128 or 255.255.255.248. Your network administrator allocates the subnet mask number to you. Together with the IP address, this number determines which network your PC and your node belongs to.

The recipient node, which is located on a subnet initially, calculates the correct network number from its own IP address and the subnet mask.
Only then does it check the node number and delivers the entire packet frame, if it corresponds.

Example of an IP address from a class B network:

IP address:	172.16.233.200	10101100 00010000 11101001 11001000
Subnet mask:	255.255.255.128	11111111 11111111 11111111 10000000
Net-ID:	172.16.00	10101100 00010000 00000000 00000000
Subnet-ID:	0.0.233.128	00000000 00000000 11101001 10000000
Host-ID:	0.0.0.72	00000000 00000000 00000000 01001000

Attention
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.

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.

IP-Header	**IP-Data**

Fig. 4-11: IP Packet

The most important information in the IP header is the IP address of the transmitter and the receiver and the transport protocol used.

4.1.3.2.2.1 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.

4.1.3.2.2.2 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 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.

4.1.3.2.3 TCP 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 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.

TCP segment

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.

4.1.3.2.4 UDP

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.

4.1.3.2.5 ARP

ARP (Address Resolution Protocol).
This protocol combines the IP address with the physical MAC address of the respective ETHERNET card. It is always used when data transfer to an IP address takes place in the same logical network in which the sender is located.

4.1.3.3 Administration and Diagnosis Protocols

In addition to the communication protocols described above, various fieldbus specific application protocols and a view protocols for system administration and diagnosis can be implemented.

- BootP
- HTTP
- DHCP
- DNS
- SNTP
- FTP
- SMTP

Additional Information
You can find a list of the exact available implemented protocols in the chapter "Technical Data" to the fieldbus coupler and/or controller.

4.1.3.3.1 BootP (Bootstrap Protocol)

The BootP protocol defines a request/response mechanism with which the MAC-ID of a fieldbus node can be assigned a fix IP address.
For this a network node is enabled to send requests into the network and call up the required network information, such as the IP address of a BootP server. The BootP server waits for BootP requests and generates the response from a configuration database.

The dynamic configuration of the IP address via a BootP server offers the user a flexible and simple design of his network. The WAGO BootP server allows any IP address to be easily assigned for the WAGO (programmable) fieldbus coupler or controller. You can download a free copy of the WAGO BootP server over the Internet at: http://www.wago.com.

Additional Information
The procedure for address allocation with the WAGO BootP Server is described in detail in the Chapter “Starting up a Fieldbus Node”.

The BOOTP Client allows for dynamic configuring of the network parameters:

Parameter	Meaning
IP address of the client	Network address of the (programmable) fieldbus coupler or controller
IP address of the router	If communication is to take place outside of the local network, the IP address of the routers (gateway) is indicated in this parameter.
Subnet mask	The Subnet mask makes the (programmable) fieldbus coupler or controller able to differentiate, which parts of the IP address determine the network and which the network station.
IP addresses of the DNS servers	Here the IP addresses can be entered by maximally 2 DNS servers.
Host name	Name of the host

When using the bootstrap protocol for configuring the node, the network parameters (IP address, etc...) are stored in the EEPROM.

Note
The network configuration is only stored in the EEPROM when the BootP protocol is used, although not if configuration is done via DHCP.

The BootP protocol is activated in the (programmable) fieldbus coupler or controller by default.

When the BootP protocol is activated, the (programmable) fieldbus coupler or controller expects a BootP server to be permanently present.
If, however, there is no BootP server available after a power-on reset, the network remains inactive.
To operate the (programmable) fieldbus coupler or controller with the IP configuration stored in the EEPROM, you must first deactivate the BootP protocol.
This is done via the web-based management system on the appropriate HTML page saved in the (programmable) fieldbus coupler or controller, which is accessed via the "Port" link.
If the BootP protocol is deactivated, the (programmable) fieldbus coupler or controller uses the parameters stored in the EEPROM at the next boot cycle.

If there is an error in the stored parameters, a blink code is output via the IO LED and configuration via BootP is automatically switched on.

4.1.3.3.2 HTTP (Hyper Text Transfer Protocol)

HTTP is a protocol used by WWW (World Wide Web) servers for the forwarding of hypermedia, texts, images, audio data, 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**.

4.1.3.3.3 DHCP (Dynamic Host Configuration Protocol)

The coupler's/controller's built-in HTML pages provide an option for IP configuration from a DHCP server, a BootP server, or the data stored in its EEPROM by default.

Note
The network configuration via DHCP is not stored in the EEPROM, this only occurs when using the BootP protocol.

The DHCP client allows dynamic network configuration of the coupler/controller by setting the following parameters:

Parameter	Meaning
IP address of the client	Network address of the coupler/controller
IP address of the router	If communication is to take place outside of the local network, the IP address of the routers (gateway) is indicated in this parameter.
Subnet mask	The Subnet mask makes the coupler/controller able to differentiate, which parts of the IP address determine the network and which the network station.
IP addresses of the DNS servers	Here the IP addresses can be entered by maximally 2 DNS servers.
Lease time	Here the maximum duration can be defined, how long the coupler/controller keeps the assigned IP address. The maximum lease time is 24.8 days. This results from the internal resolution of timer.
Renewing time	The Renewing time indicates, starting from when the coupler/controller must worry about the renewal of the leasing time.
Rebinding time	The Rebinding time indicates, after which time the coupler/controller must have gotten its new address.

In the case of configuration of network parameters via the DHCP protocol, the coupler/controller automatically sends a request to a DHCP server after initialization. If there is no response, the request is sent again after 4 seconds, a further one after 8 seconds and again after 16 seconds. If all requests remain unanswered, a blink code is output via the “IO” LED. Transfer of the parameters from the EEPROM is not possible.

Where a lease time is used, the values for the renewing and rebinding time must also be specified. After the renewing time expires, the coupler/controller attempts to automatically renew the lease time for its IP address . If this continually fails up to the rebinding time, the coupler/controller attempts to obtain a new IP address. The time for the renewing should be about one half of the lease time. The rebinding time should be about $^7/_8$ of the lease time.

4.1.3.3.4 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 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.

4.1.3.3.5 SNTP-Client (Simple Network Time Protocol)

The SNTP client is used for synchronization of the time of day between a time server (NTP and SNTP server Version 3 and 4 are supported) and the clock module integrated in the (programmable) fieldbus coupler or controller. The protocol is executed via a UDP port. Only unicast addressing is supported.

Configuration of the SNTP client

The configuration of the SNTP client is performed via the web-based management system under the “Clock” link. The following parameters must be set:

Parameter	Meaning
Address of the Time server	The address assignment can be made either over a IP address or a host name.
Time zone	The time zone relative to GMT (Greenwich Mean time). A range of -12 to +12 hours is acceptable.
Update Time	The update time indicates the interval in seconds, in which the synchronization with the time server is to take place.
Enable Time Client	It indicates whether the SNTP Client is to be activated or deactivated.

4.1.3.3.6 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 1.5 MB is available for the file system. The file system is mapped to RAM disk. To permanently store the data of the RAM disk, the information is additionally copied into the flash memory. The data is stored in the flash after the file has been closed. Due to the storage process, access times during write cycles are long.

Attention
Up to 1 million write cycles are possible for writing to the flash memory for the file system.

The following table shows the supported FTP commands for accesses to the file system:

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 RNTO)
RNTO	Renames file in (with RNFR)
ABOR	Stops current function
DELE	Deletes file
CWD	Changes directory
LIST	Gives the directory list

Command	Function
NLST	Gives the directory list
RMD	Deletes directory
PWD	Gives the actually path
MKD	Puts on a directory

The TFTP (Trivial File Transfer Protocol) is not supported by some of the couplers/controllers.

Additional Information
You can find a list of the exact available implemented protocols in the chapter "Technical Data" to the fieldbus coupler and/or controller.

4.1.3.3.7 SMTP (Simple Mail Transfer Protocol)

The Simple Mail Transfer Protocol (SMTP) enables sending of ASCII text messages to mail boxes on TCP/IP hosts in a network. It is therefore used for sending and receiving e-mails.

The e-mail to be sent is created with a suitable editor and placed in a mail out basket.

A send SMTP process polls the out-basket at regular intervals and therefore finds mail waiting to be sent. It then establishes a TCP/IP connection with the target host, to which the message is transmitted. The receive SMTP process on the target host accepts the TCP connection. The message is then transmitted and finally placed in an in-basket on the target system. SMTP expects the target system to be online, otherwise no TCP connection can be established. Since many desktop computers are switched off at the end of the day, it is impractical to send SMTP mail there. For that reason, in many networks special SMTP hosts are installed in many networks, which are permanently switched on to enable distribution of received mail to the desktop computers.

4.1.3.4 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. There are based on ETHERNET couplers/controllers available developed by WAGO, with the following possible application protocols:

- MODBUS TCP (UDP)
- EtherNet/IP
- BACnet/IP
- KNXnet/IP
- PROFINET
- Powerlink
- SERCOS III

Additional Information
You can find a list of the exact available implemented protocols in the chapter "Technical Data" to the fieldbus coupler and/or controller.

If fieldbus specific application protocols are implemented, then these protocols are individual described in the following chapters.

4.2 BACnet/IP

4.2.1 General

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 BACnet network model works independently of the system. When using devices, data/information are 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.

4.2.1.1 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.

4.2.1.2 BACnet Components

According to the BACnet Standard, 25 different objects and 38 services are supported (Last update: ANSI/ASHRAE 135-2004, DIN EN ISO 16484-5). An object is composed of several object-specific properties. Services are also present and transmit their data in certain structures.

4.2.1.2.1 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 25 objects are tailored specifically to the HVAC sector for building automation services. They contain both physical inputs and outputs and virtual objects.

Tab. 4-36: BACnet objects

Object		Function
1	Accumulator	Count value entry; counts values
2	Analog Input	Analog inputs, measures voltage, temperature, etc.
3	Analog Output	Analog outputs, e.g. for setting controllers
4	Analog Value	Analog value; e.g. from a calculation
5	Averaging	Average for calculations and statistics
6	Binary Input	Binary input, reports an event or disturbance
7	Binary Output	Binary output; alternates between two states
8	Binary Value	Binary value, virtual binary data time
9	Calendar	Operating calendar, list of holidays and vacations
10	Command	(Group) request to execute predefined activities (e.g. a list of switching commands)
11	Device	Device, information on the respective BACnet device
12	Event Enrollment	Event category, defined reactions to events
13	File	File, transmits data
14	Group	Group entry, grouped object values in the device
15	Life-Safety-Point	Hazard reporter, delivers information on properties with regard to notifications of danger
16	Life-Safety-Zone	Security area, includes hazard reporting objects according to certain criteria
17	Loop	Controller, performs regulation functions
18	Multi-State Input	Multi-level input, delivers reports on states such as off/on, open/closed as a coded number
19	Multi-State Output	Multi-level output, delivers the output states of commands

Object	Function
20 Multi-State Value	Multi-level value, delivers logical states of the object
21 Notification Class	Notification class, assigns alarm and event reports to times and recipients
22 Program	Program, used to access programs in the BACnet device
23 Pulse Converter	Impulse entry, counts quantities for control and monitoring
24 Schedule	Schedule, for establishing certain actions at predefined times
25 Trend Log	Trend log of properties of certain criteria, either cyclic or after value changes (COV)

4.2.1.2.2 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

Other properties are object-specific. Currently, there are over 150 different properties (BACnet Standard 135-2004).

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.

4.2.1.2.3 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.

Thirty-eight (38) standardized services are divided into 5 categories (Last update: 2005)

- Alarms and events (Alarm and Event Services)
- Object access (Object Access Services)
- File access (File Access Services)
- Remote device access (Remote Device Management Service)
- Virtual terminal services (Virtual Terminal Service)

Alarms and events (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

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 (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
- **ReadPropertyConditional Service**
  Reads the content of several objects properties according to special criteria
- **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 (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, written and closed again, as a whole or in part.

Remote device access (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 services (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

4.2.1.2.3.1 Client-Server Communication

In Client-Server communication, Clients and Servers are considered to have different functions. A Clients 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.

4.2.1.2.3.2 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 Fig. 4-1). The use on the highest priority level with the lowest number has priority.

Tab. 4-37: Priority levels

Priority Level	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.

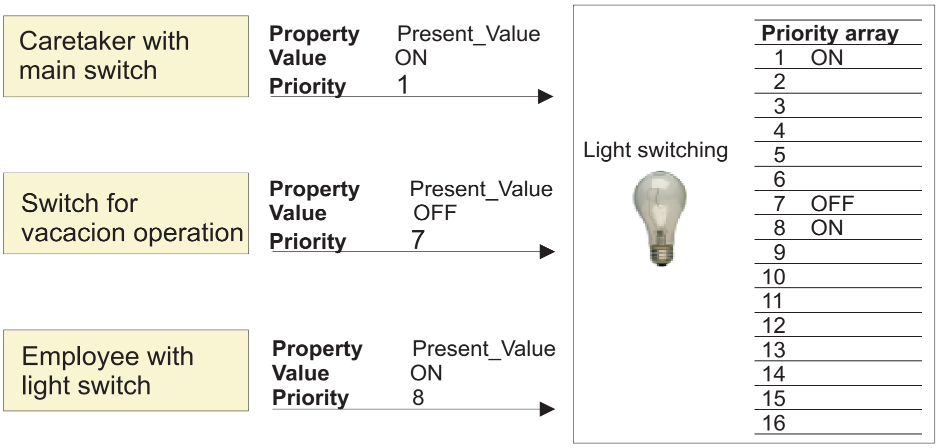


Fig. 4-1: Examples of a prioritization

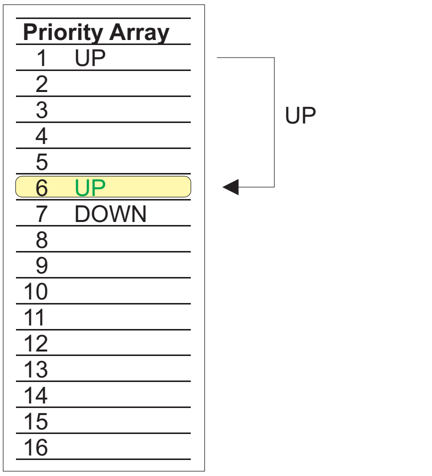
G083015e1-4

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.

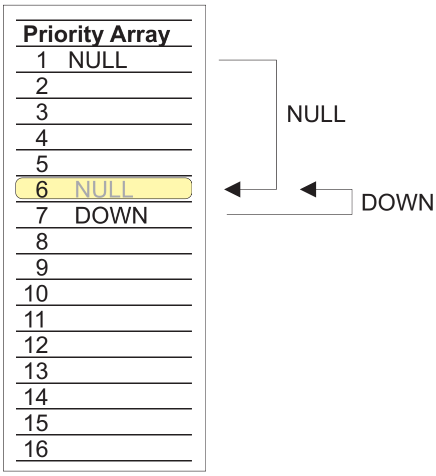
Example:

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.



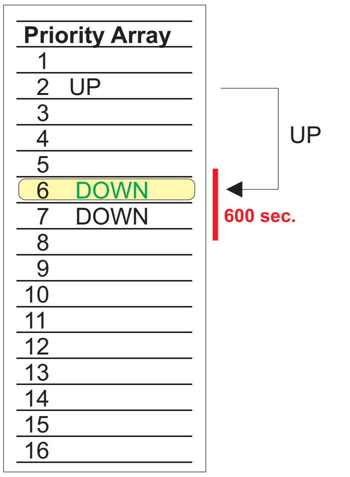
The change to the value "UP", with priority "1", has precedence and is registered as the Present_Value in the object.

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.



If, after some time, the priority level "1" is overwritten with "ZERO", the status of the minimum on/off switching time is also overwritten with ZERO and reset.

The value in the next higher priority level "7" is taken over as the "Present_Value" with "DOWN".



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 ZERO must be resent. If all priority levels are assigned the value ZERO, the standard value will be assumed.

4.2.1.2.4 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).

4.2.1.2.4.1 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 (2) 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 **D**ata **S**haring-**R**ead**P**roperty-**A**
The Client (A) makes the request to read a value (RP), to the Server (B).

4.2.1.2.4.2 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.

4.2.1.2.4.3 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.

4.2.1.2.4.4 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 **T**rending-**V**iewing and **M**odifying **T**rends-**A**
Device A displays the trending records of Device B and alters the Trend-Log Parameter there.

4.2.1.2.4.5 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 **D**evice **M**anagement **D**ynamic **D**evice **B**inding-**A**
Device A obtains information on other devices within the network and interprets device information.

4.2.1.2.4.6 Network Management (NM)

NM regulates terminal sessions and remote uses. Connections to other devices are established.

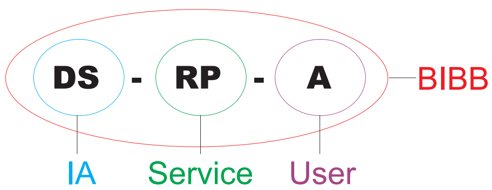
Example: NM-CE-B **N**etwork **M**anagement **C**onnection **E**stablishment-**B**
Device B executes commands to create and close connections.

Each IA is composed of several BIBBs. These contain functions that vary depending on device profile (see section 4.2.1.2.7).

4.2.1.2.5 BACnet Interoperability Building Blocks (BIBB)s

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:



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).

4.2.1.2.6 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.

4.2.1.2.7 Device Profile

There are six 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 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

Additional Information
In addition to these device profiles, there is an additional profile with the BACnet Gateway (B-GW), which has only functioned up to now as a suggestion of the B.I.G. EU and is not anchored in the BACnet Standard (last updated: ASHRAE Standard 135-2004, Version 1, Revision 4).

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 Tab. 4-38) and makes devices comparable with each other.

Tab. 4-38: IAs/BIBBs of the six standardized device profiles

IA	B-OWS	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-RP-A,B DS-RPM-A,B DS-WP-A,B DS-WPM-B DS-COVU-A,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 Mgmt	AE-N-A AE-ACK-A AE-INFO-A AE-ESUM-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-A	SCHED-E-B	SCHED-I-B			
Trending	T-VMT-A T-ATR-A	T-VMT-I-B T-ATR-B				
Device & Network Mgmt	DM-DDB-A,B DM-DOB-A,B DM-DCC-A DM-TS-A DM-UTC-A DM-RD-A DM-BR-A NM-CE-A	DM-DDB-A,B DM-DOB-A,B DM-DCC-B DM-TS-B or DM-UTC-B DM-RD-B DM-BR-B NM-CE-A	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		

4.2.1.2.7.1 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.

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

4.2.1.2.8 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).

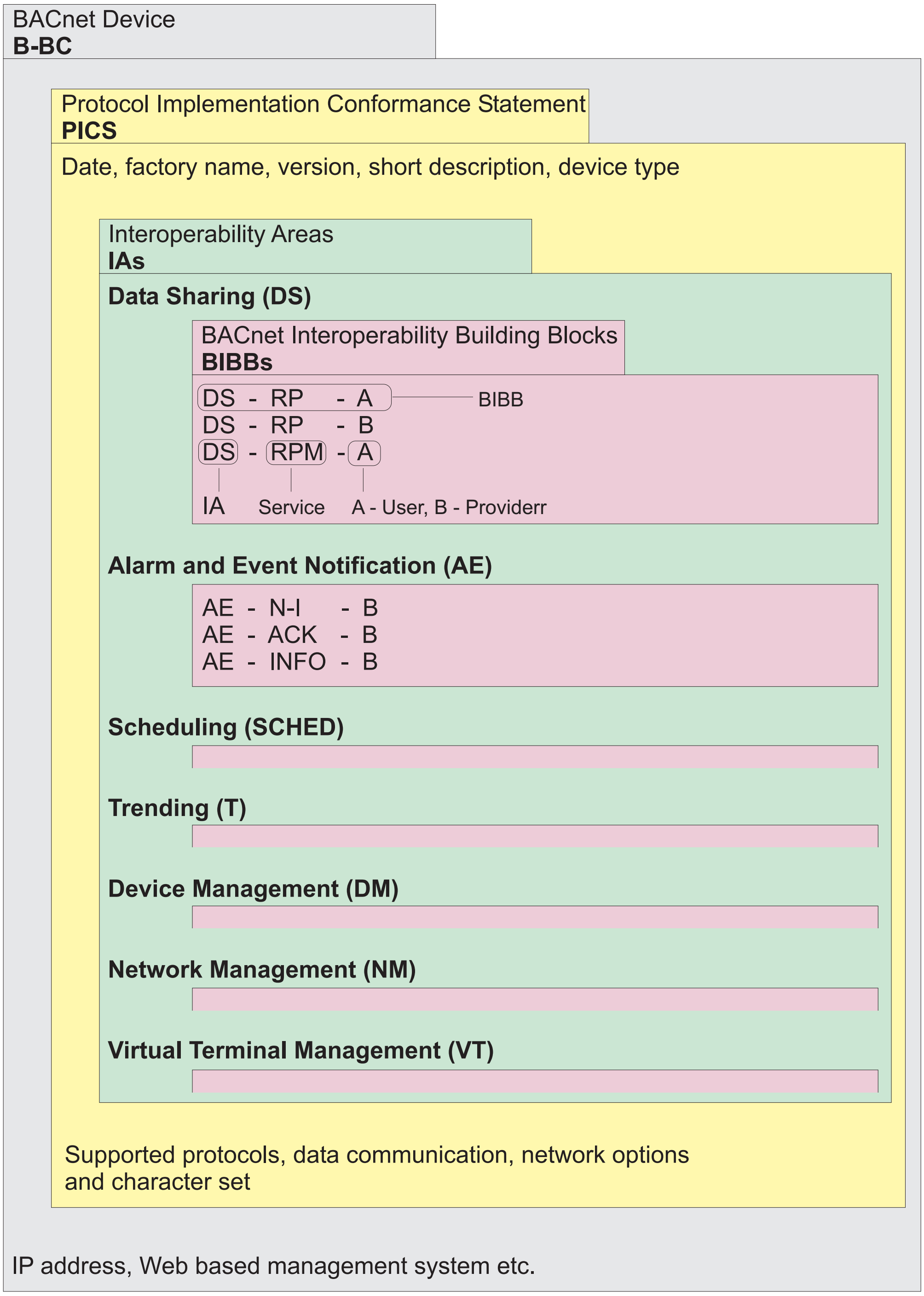


Fig. 4-12 Connection between BACnet Components G083012d

4.2.1.3 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.

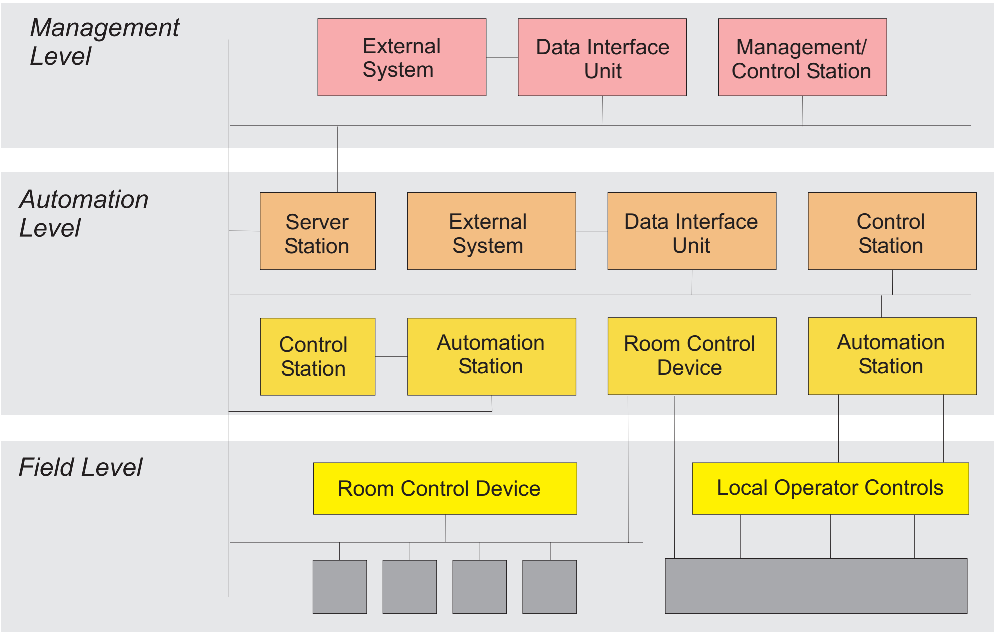


Fig. 4-13 Use of BACnet in building automation

4.2.1.4 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 4.2.1.2.4 and 4.2.1.2.5).

Application notes

- Facility Management
- Finance and Personnel Planning
- Energy Management
- Servicing

4.2.1.4.1 Automation Level

The automation level includes alarm organization and the operation of systems.

Areas of use

- Monitoring
- Optimizing
- Control
- Regulating
- Reporting

4.2.1.4.2 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

4.2.1.5 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.

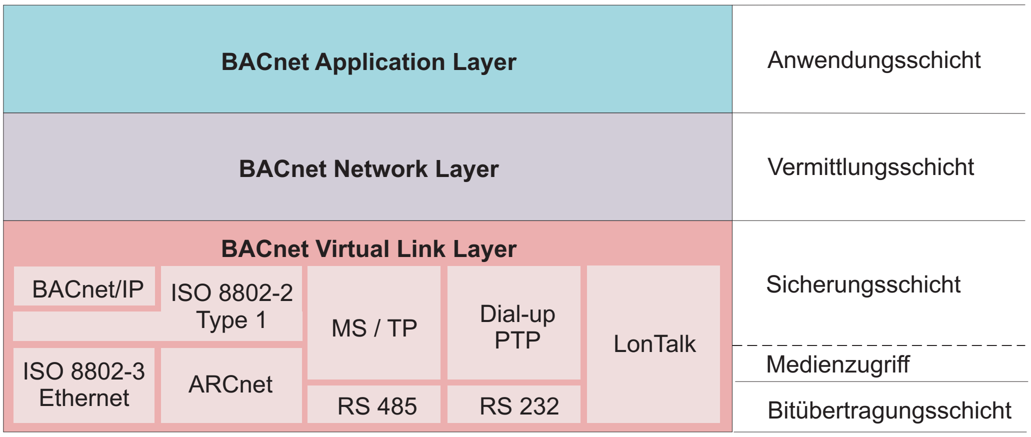


Fig. 4-14: BACnet layers in the ISO/OSI Model G083006d

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 RS232** as serial connections; enables point-to-point connections between two participants, e.g. over a telephone line.
- **MS/TP (Master-Slave/Token-Passing) via RS485** 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)

4.2.1.6 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

4.2.1.6.1 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 Fig. 4-2).

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.

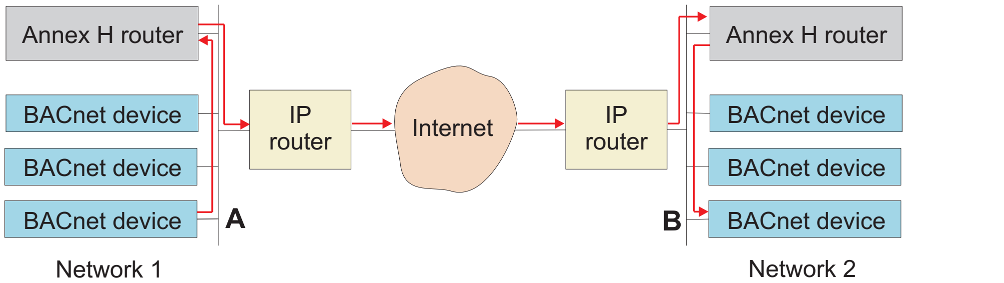


Fig. 4-2: Communication over an Annex-H router

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.

4.2.1.6.2 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 Fig. 4-3).

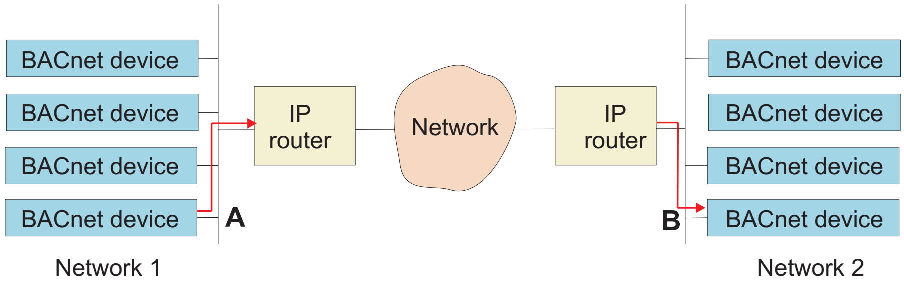


Fig. 4-3: Communication via BACnet/IP G083010e

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.

4.2.1.6.2.1 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 Fig. 4-4).

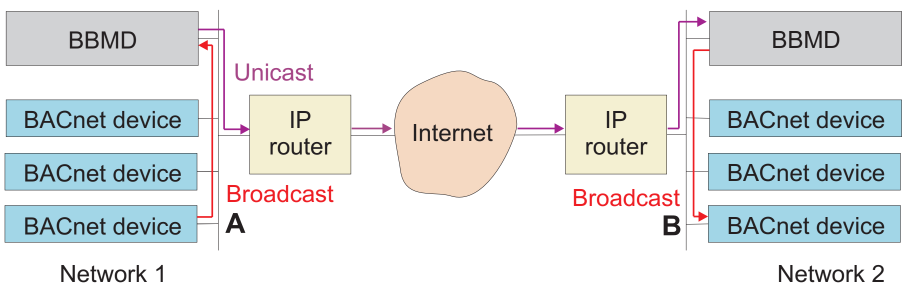


Fig. 4-4: Sending a broadcast report

G083017e

4.2.1.6.2.2 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 (see Fig. 4-4).

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 Fig. 4-5). 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 networks.

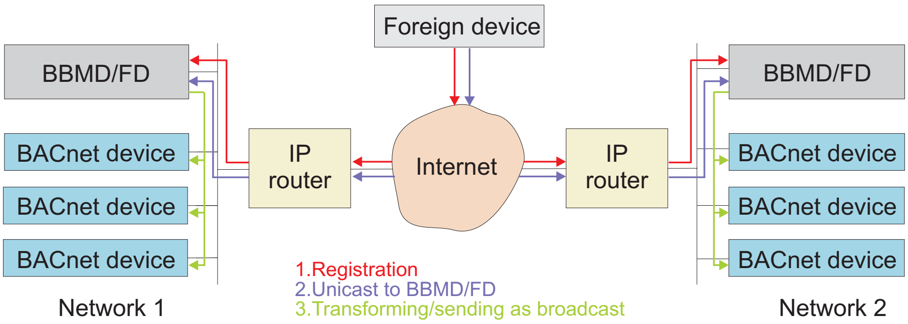


Fig. 4-5: Registration of a foreign device G083011d

4.3 MODBUS Functions

4.3.1 General

MODBUS is a manufacturer-independent, open fieldbus standard for diverse applications in manufacturing and process automation.

The MODBUS protocol is implemented for the transmission of the process image, the fieldbus variables, different settings and information on the controller according to the current Internet Draft of the IETF (Internet Engineering Task Force).
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:

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

Fig. 4-15: MODBUS/TCP Header

Additional Information
The structure of a datagram is specific for the individual function. Refer to the descriptions of the MODBUS Function Codes.

For the MODBUS protocol 15 connections are made available over TCP. Thus it allows digital and analog output data to be directly read out at a fieldbus node and special functions to be executed by way of simple MODBUS function codes from 15 stations simultaneously.
For this purpose a set of MODBUS functions from the *OPEN MODBUS /TCP SPECIFICATION* is realized.

Additional 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:

Data type	Length	Description
Discrete Inputs	1 Bit	Digital Inputs
Coils	1 Bit	Digital Outputs
Input Register	16 Bit	Analog-Input data
Holding Register	16 Bit	Analog-Output data

For each basic data type one or more „Function codes“ are defined.

These functions allow digital or analog input and output data, and internal variables to be set or directly read out of the fieldbus node.

Function code	hexadec.	Function	Access method and description	Access to resources
FC1:	0x01	Read Coils	Reading of several single input bits	R: Process image, PFC variables
FC2:	0x02	Read Input Discretes	Reading of several input bits	R: Process image, PFC variables
FC3:	0x03	Read Multiple Registers	Reading of several input registers	R: Process image, PFC variables, internal variables, NOVRAM
FC4:	0x04	Read Input Registers	Reading of several input registers	R: Process image, PFC variables, internal variables, NOVRAM
FC5:	0x05	Write Coil	Writing of an individual output bit	W: Process image, PFC variables
FC6:	0x06	Write Single Register	Writing of an individual output register	W: Process image, PFC variables, internal variables, NOVRAM
FC 11:	0x0B	Get Comm Event Counters	Communication event counter	R: None
FC 15:	0x0F	Force Multiple Coils	Writing of several output bits	W: Process image, PFC variables
FC 16:	0x0010	Write Multiple Registers	Writing of several output registers	W: Process image, PFC variables, internal variables, NOVRAM
FC 22:	0x0016	Mask Write Register		W: Process image, PFC variables, NOVRAM
FC 23:	0x0017	Read/Write Registers	Reading and writing of several output registers	R/W: Process image, PFC variables, NOVRAM

Tab. 4-3: List of the MODBUS Functions in the Fieldbus Controller

To execute a desired function, specify the respective function code and the address of the selected input or output data.

Attention
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.

4.3.2 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.

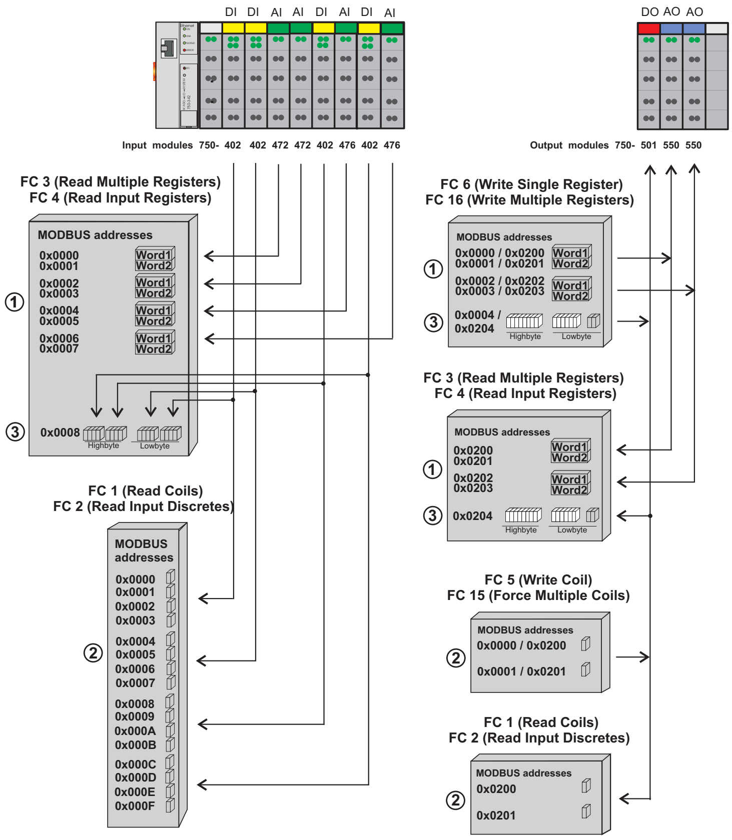


Fig. 4-16: Use of the MODBUS Functions G012918e

Attention
It is recommended that analog data be accessed with register functions (1) and digital data with coil functions (2).

4.3.3 Description of the MODBUS Functions

All MODBUS functions are executed as follows:

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. 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:

Exception Code	Meaning
0x01	Illegal Function
0x02	Illegal Data Address
0x03	Illegal Data Value
0x04	Slave Device Failure
0x05	Acknowledge
0x06	Server Busy
0x08	Memory Parity Error
0x0A	Gateway Path Unavailable
0x0B	Gateway Target Device Failed To Respond

The following chapters describe the datagram architecture of request, response and exception with examples for each function code.

Note
In the case of the read functions (FC1 – FC4) the outputs can be additionally written and read back by adding an offset of 200_{hex} (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} - $62FC_{hex}$].

4.3.3.1 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.

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).

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 with 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

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x81
Byte 8	Exception code	0x01 or 0x02

4.3.3.2 Function Code FC2 (Read Input Discretes)

This function reads the input bits from a slave device.

Request

The request specifies the reference number (starting address) and the bit count to be read.
Example: Read input bits 0 to 7:

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).

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 with 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

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x82
Byte 8	Exception code	0x01 or 0x02

4.3.3.3 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:

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.

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

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x83
Byte 8	Exception code	0x01 or 0x02

4.3.3.4 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:

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.

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

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x84
Byte 8	Exception code	0x01 or 0x02

4.3.3.5 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):

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

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x05
Byte 8, 9	Reference number	0x0001
Byte 10	Value	0xFF
Byte 11		0x00

Exception

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01, 0x02 or 0x03

4.3.3.6 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.

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.

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x06
Byte 8, 9	Reference number	0x0001
Byte 10, 11	Register Value	0x1234

Exception

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

4.3.3.7 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

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.

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

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

4.3.3.8 Function Code FC15 (Force Multiple Coils)

This function sets a sequence of output bits to 1 or 0 in a slave device. The maximum number is 256 bits.

Request

The request message specifies the reference number (first coil in the sequence), the bit count (number of bits to be written), and the output data. The output coils are zero-based; therefore, the first output point is 0.

In this example 16 bits are set, starting with the address 0. The request contains 2 bytes with the value 0xA5F0, or 1010 0101 1111 0000 in binary format.

The first data byte transmits the value of 0xA5 to the addresses 7 to 0, whereby 0 is the lowest value bit. The next byte transmits 0xF0 to the addresses 15 to 8, whereby the lowest value bit is 8.

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

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x0F
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit Count	0x0010

Exception

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x8F
Byte 8	Exception code	0x01 or 0x02

4.3.3.9 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:

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

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word Count	0x0002

Exception

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

4.3.3.10 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

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

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

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

4.3.3.11 Function Code FC23 (Read/Write multiple registers)

This function performs a combination of a read and write operation in a single request. The function can write the new data to a group registers, and then return the data of a different group.

Request

The reference numbers (addresses) are zero-based in the request message; therefore, the first register is at address 0.
The request message specifies the registers to read and write. The data is sent as 2 bytes per register. Example: The data in register 3 is set to value 0x0123, and values 0x0004 and 0x5678 are read out of the two registers 0 and 1.

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x000F
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x17
Byte 8-9	reference number for read	0x0000
Byte 10-11	Word count for read (1-125)	0x0002
Byte 12-13	reference number for write	0x0003
Byte 14-15	Word count for write (1-100)	0x0001
Byte 16	Byte Count (B = 2 x word count for write)	0x02
Byte 17-(B+16)	Register Values	0x0123

Response

Byte	Field name	Example
....		
Byte 7	MODBUS function code	0x17
Byte 8	Byte Count (B = 2 x word count for read)	0x04
Byte 9-(B+1)	Register Values	0x0004, 0x5678

Exception

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x97
Byte 8	Exception code	0x01 or 0x02

Attention
If register areas for read and write overlap, the results are undefined.

4.3.4 MODBUS Register Mapping

The following tables display the MODBUS addressing and the corresponding IEC61131 addressing for the process image, the PFC variables, the NOVRAM 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 (Word) Access Reading (with FC3, FC4 and FC23):

MODBUS-Address [dec]	[hex]	IEC61131 Address	Memory Range
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 4.3.5.3 Configuration Functions)
12288 ... 24575	0x3000 ... 0x5FFF	%MW0 ... %MW12287	NOVRAM 8kB retain memory (max. 24kB)
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 ... 29436	0x7000 ... 0x72FC	%QW512 ... %QW1275	Physical Output Area (2) Additional 764 Words physical output data
29437 ... 65535	0x72FD ... 0xFFFF	-	MODBUS Exception: “ Illegal data address”

Register (Word) Access Writing (with FC6, FC16, FC22 and FC23):

MODBUS-Address [dec]	[hex]	IEC61131 Address	Memory Range
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 4.3.5.3 Configuration Functions)

MODBUS-Address [dec]	[hex]	IEC61131 Address	Memory Range
12288 ... 24575	0x3000 ... 0x5FFF	%MW0 ... %MW12287	NOVRAM 8kB retain memory (max. 24kB)
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 ... 29436	0x7000 ... 0x72FC	%QW512 ... %QW1275	Physical Output Area (2) Additional 764 Words physical output data
29437 ... 65535	0x72FD ... 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):

MODBUS Address [dec]	[hex]	Memory Range	Description
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 8kB retain memory (max. 24kB)
32768 ... 34295	0x8000 ... 0x85F7	Physical Input Area (2)	Starts with the 513th and ends with the 2039th digital input
34296 ... 36863	0x85F8 ... 0x8FFF		MODBUS Exception: "Illegal data address"
36864 ... 38391	0x9000 ... 0x95F7	Physical Output Area (2)	Starts with the 513th and ends with the 2039th digital output
38392 ... 65535	0x95F8 ... 0xFFFF		MODBUS Exception: "Illegal data address"

Bit Access Writing (with FC5 and FC15):

MODBUS Address [dec]	MODBUS Address [hex]	Memory Range	Description
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 8kB retain memory (max. 24kB)
32768 ... 34295	0x8000 ... 0x85F7	Physical Output Area (2)	Starts with the 513th and ends with the 2039th digital output
34296 ... 36863	0x85F8 ... 0x8FFF		MODBUS Exception: “Illegal data address”
36864 ... 38391	0x9000 ... 0x95F7	Physical Output Area (2)	Starts with the 513th and ends with the 2039th digital output
38392 ... 65535	0x95F8 ... 0xFFFF		MODBUS Exception: “Illegal data address”

4.3.5 Internal Variables

Address	Access	Length (word)	Remark
0x1000	R/W	1	Watchdog-Time read/write
0x1001	R/W	1	Watchdog Coding mask 1-16
0x1002	R/W	1	Watchdog Coding mask 17-32
0x1003	R/W	1	Watchdog Trigger
0x1004	R	1	Minimum Trigger time
0x1005	R/W	1	Watchdog stop (Write sequence 0xAAAA, 0x5555)
0x1006	R	1	Watchdog Status
0x1007	R/W	1	Restart Watchdog (Write sequence 0x1)
0x1008	RW	1	Stop Watchdog (Write sequence 0x55AA or 0xAA55)
0x1009	R/W	1	MODBUS -and HTTP- close at Watchdog Timeout
0x100A	R/W	1	Watchdog configuration
0x100B	W	1	Save Watchdog parameter
0x1020	R	1-2	LED Error Code
0x1021	R	1	LED Error Argument
0x1022	R	1-4	Number of analog output data in the process image (in bits)
0x1023	R	1-3	Number of analog input data in the process image (in bits)
0x1024	R	1-2	Number of digital output data in the process image (in bits)
0x1025	R	1	Number of digital input data in the process image (in bits)
0x1028	R/W	1	Boot configuration

Address	Access	Length (word)	Remark
0x1029	R	9	MODBUS-TCP statistics
0x102A	R	1	Number of TCP connections
0x1030	R/W	1	Configuration MODBUS/TCP Timeout
0x1031	W	1	Read out the MAC-ID of the controller
0x1050	R	3	Diagnosis of the connected I/O Modules
0x2000	R	1	Constant 0x0000
0x2001	R	1	Constant 0xFFFF
0x2002	R	1	Constant 0x1234
0x2003	R	1	Constant 0xAAAA
0x2004	R	1	Constant 0x5555
0x2005	R	1	Constant 0x7FFF
0x2006	R	1	Constant 0x8000
0x2007	R	1	Constant 0x3FFF
0x2008	R	1	Constant 0x4000
0x2010	R	1	Firmware version
0x2011	R	1	Series code
0x2012	R	1	Controller code
0x2013	R	1	Firmware versions major revision
0x2014	R	1	Firmware versions minor revision
0x2020	R	16	Short description controller
0x2021	R	8	Compile time of the firmware
0x2022	R	8	Compile date of the firmware
0x2023	R	32	Indication of the firmware loader
0x2030	R	65	Description of the connected busmodules (module 0–64)
0x2031	R	64	Description of the connected busmodules (module 65-128)
0x2032	R	64	Description of the connected busmodules (module 129-192)
0x2033	R	63	Description of the connected busmodules (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

4.3.5.1 Description of the internal variables

4.3.5.1.1 Watchdog (Fieldbus failure)

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 use 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. Finally, the Watchdog-Trigger register (0x1003) must be changed to a non-zero value to start the timer.

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 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)

4.3.5.1.2 Watchdog Register

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.

Register address 0x1000 (MODBUS Address 404097)	
Designation	Watchdog time, WS_TIME
Access	read / write
Default	0x0000
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.

Register address 0x1001 (MODBUS Address 404098)	
Designation	Watchdog function coding mask, function code 1...16, WDFCM_1_161...16
Access	read / write
Default	0x0000
Description	Using this mask, the specific function codes can be configured to reset the watchdog function. The function code can be selected by writing a '1' to the appropriate bit(s) (2 (Function code-1) +2 (Function code-1...). Bit 1001.0 corresponds to function code1, Bit 1001.1 corresponds to function code2... A value of 0xFF enables MODBUS functions code 1 through 16 to reset the watchdog. It is not possible to modify this value while the watchdog is running.

Register address 0x1002 (MODBUS Address 404099)	
Designation	Watchdog function coding mask, function code 17...32, WD_FCM_17_32
Access	read / write
Default	0x0000
Description	Same function as above, however, with the function codes 17 to 32. These codes are currently not supported, for this reason the default value should not be changed.. It is not possible to modify this value while the watchdog is running.

Register address 0x1003 (MODBUS Address 404100)	
Designation	Watchdog-Trigger, WD_TRIGGER
Access	read / write
Default	0x0000
Description	This register is used to trigger the watchdog. The default value after power up is 0. The writing of a non zero value will trigger the watchdog. The watchdog is triggered each time the contents of this register are modified. The watchdog cannot be triggered if the watchdog timer register is set to 0.

Register address 0x1004 (MODBUS Address 404101)	
Designation	Minimum current trigger time, WD_AC_TRG_TIME
Access	read / write
Default	0xFFFF
Description	This register stores the time value for the shortest remaining watchdog duration. The default value is 0xFFFF. When the watchdog timer is triggered, this register is continuously compared to the remaining watchdog time, and the lesser of the two values is stored in this register. If the value in this register is 0, a watchdog fault has occurred.

Register address 0x1005 (MODBUS Address 404102)	
Designation	Stop Watchdog, WD_AC_STOP_MASK
Access	read / write
Default	0x0000
Description	This register is used to stop the watchdog timer by entering a value of 0xAAAA followed by 0x5555.

Register address 0x1006 (MODBUS Address 404103)	
Designation	While watchdog is running, WD_RUNNING
Access	read
Default	0x0000
Description	Current watchdog status. at 0x0000: Watchdog not active, at 0x0001: Watchdog active. at 0x0002: Watchdog exhausted.

Register address 0x1007 (MODBUS Address 404104)	
Designation	Restart watchdog, WD_RESTART
Access	read / write
Default	0x0001
Description	This register restarts the watchdog timer by writing a value of 0x1 into it. If the watchdog was stopped before the overrun, it is not restarted.

Register address 0x1008 (MODBUS Address 404105)	
Designation	Simply stop watchdog WD_AC_STOP_SIMPLE
Access	read / write
Default	0x0000
Description	This register stops the watchdog by writing the value 0x0AA55 or 0X55AA into it. The watchdog timeout fault is deactivated and it is possible to write in the watchdog register again. If there is an existing watchdog fault, it is reset

Register address 0x1009 (MODBUS Address 404106)	
Designation	Close MODBUS socket after watchdog timeout
Access	read / write
Description	0: MODBUS socket is not closed 1: MODBUS socket is closed

Register address 0x100A (MODBUS Address 404107)	
Designation	Alternative watchdog
Access	read / write
Default	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 watch dog times out, all outputs are set to zero. The outputs will become operational again, after communications are re-established.

All register data is in word format.

Examples:

Set the watchdog for a timeout of 1 second. Function code 5 (Force Single Coil) will be use to reset the watchdog time.

1. Write 0x000A (1000ms /100 ms) in the Watchdog Timer register (0x1000).
2. Write 0x0010 ($2^{(5-1)}$) in the Coding Mask register (0x1001)
3. Modify the value of the Watchdog-Trigger register (0x0003) to start the watchdog.
4. At this point, the fieldbus master must continuously use function code 5 (Force Single Coil) within the specified time to reset the watchdog timer. If time between requests exceeds 1 second, a watchdog timeout error occurs.

To stop the watchdog after it is started, write the value 0x0AA55 or 0X55AA into it the Simply Stop Watchdog register (0x1008).

Set the watchdog for a timeout of 10 minutes. Function code 3 (Read Multiple Registers) will be use to reset the watchdog time.

1. Write 0x1770 (10*60*1000 ms / 100 ms) in the register for time overrun (0x1000).
2. Write 0x0004 ($2^{(3-1)}$) in the Coding Mask register (0x1001)
3. Modify the value of the Watchdog-Trigger register (0x0003) to start the watchdog.
4. At this point, the fieldbus master must continuously use function code 3 (Force Single Coil) within the specified time to reset the watchdog timer. If time between requests exceeds 10 minutes, a watchdog timeout error occurs..

To stop the watchdog after it is started, write the value 0x0AA55 or 0X55AA into it the Simply Stop Watchdog register (0x1008).

Register address 0x100B	
Value	Save Watchdog Parameter
Access	write
Default	0x0000
Description	With writing of '1' in register 0x100B the registers 0x1000, 0x1001, 0x1002 are set on remanent.

4.3.5.2 Diagnostic Functions

The following registers can be read to determine errors in the node:

Register address 0x1020 (MODBUS Address 404129)	
Designation	LedErrCode
Access	read
Description	Declaration of the Error code (see section 3.1.8.4 for error code definitions)

Register address 0x1021 (MODBUS Address 404130)	
Designation	LedErrArg
Access	read
Description	Declaration of the Error argument (see section 3.1.8.4 for error code definitions)

4.3.5.3 Configuration Functions

The following registers contain configuration information of the connected modules:

Register address 0x1022 (MODBUS Address 404131)	
Designation	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)

Register address 0x1023 (MODBUS Address 404132)	
Designation	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)

Register address 0x1024 (MODBUS Address 404133)	
Designation	CnfLen.DigitalOut
Access	read
Description	Number of digital output bits in the process image

Register address 0x1025 (MODBUS Address 404134)	
Designation	CnfLen.DigitalInp
Access	read
Description	Number of digital input bits in the process image

Register address 0x1028 (MODBUS Address 404137)	
Designation	Boot options
Access	read / write
Description	Boot configuration: 1: BootP 2: DHCP 4: EEPROM

Register address 0x1029 (MODBUS Address 404138, with 9 words)	
Designation	MODBUS TCP statistics
Access	read / write
Description	1 word SlaveDeviceFailure → internal bus error, F-bus 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.

Register address 0x102A (MODBUS Address 404139, with a word count of 1)	
Designation	MODBUS/TCP Connections
Access	read
Description	Number of TCP connections

Register address 0x1030 (MODBUS Address 404145, with a word count of 1)	
Designation	Configuration MODBUS/TCP Timeout
Access	read / write
Default	0x0000
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 timeout, idle connection will be closed. Outputs remain in last state. Default value is 0 (timeout disabled).

Register address 0x1031 (MODBUS Address 404146, with a word count of 3)	
Designation	Read the MAC-ID of the controller
Access	read
Description	This register gives the MAC-ID, with a length of 3 words

Register address 0x1050 (MODBUS Address 404177, with a word count of 3) since Firmware version 9	
Designation	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

Register address 0x2030 (MODBUS Address 408241, with a word count of up to 65)	
Designation	Description of the connected I/O modules
Access	Read modules 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 item 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			

Register address 0x2031 (MODBUS Address 408242, with a word count of up to 64)	
Designation	Description of the connected busmodules
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 item 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

Register address 0x2032 (MODBUS Address 408243, with a word count of up to 64)	
Designation	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 item 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

Register address 0x2033 (MODBUS Address 408244, with a word count of up to 63)	
Designation	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 item 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

Register address 0x2040 (MODBUS Address 408257)	
Designation	Implement a software reset
Access	write (Write sequence 0xAA55 or 0x55AA)
Description	With Writing 0xAA55 or 0x55AA the register will be reset.

Register address 0x2041 (MODBUS Address 408258) since Firmware version 3	
Designation	Flash Format
Access	write (Write sequence 0xAA55 or 0x55AA)
Description	The file system Flash is again formatted.

Register address 0x2042 (MODBUS Address 408259) since Firmware version 3	
Designation	Extract data files
Access	write (Write sequence 0xAA55 or 0x55AA)
Description	The standard files (HTML pages) of the Controller are extracted and written into the Flash.

Register address 0x2043 since Firmware version 9	
Designation	0x55AA
Access	write
Description	Factory Settings

4.3.5.4 Firmware Information

The following registers contain information on the firmware of the controller:

Register address 0x2010 (MODBUS Address 408209, with a word count of 1)	
Designation	Revision, INFO_REVISION
Access	Read
Description	Firmware Index, e. g. 0005 for version 5

Register address 0x2011 (MODBUS Address 408210, with a word count of 1)	
Value	Series code, INFO_SERIES
Access	Read
Description	WAGO serial number, e. g. 0750 for WAGO-I/O-SYSTEM 750

Register address 0x2012 (MODBUS Address 408211, with a word count of 1)	
Value	Item number, INFO_ITEM
Access	Read
Description	WAGO item number, e. g. 841 for the controller

Register address 0x2013 (MODBUS Address 408212, with a word count of 1)	
Value	Major sub item code, INFO_MAJOR
Access	read
Description	Firmware version Major Revision

Register address 0x2014 (MODBUS Address 408213, with a word count of 1)	
Value	Minor sub item code, INFO_MINOR
Access	read
Description	Firmware version Minor Revision

Register address 0x2020 (MODBUS Address 408225, with a word count of up to 16)	
Value	Description, INFO_DESCRIPTION
Access	Read
Description	Information on the controller, 16 words

Register address 0x2021 (MODBUS Address 408226, with a word count of up to 8)	
Value	Description, INFO_DESCRIPTION
Access	Read
Description	Time of the firmware version, 8 words

Register address 0x2022 (MODBUS Address 408227, with a word count of up to 8)	
Value	Description, INFO_DATE
Access	Read
Description	Date of the firmware version, 8 words

Register address 0x2023 (MODBUS Address 408228, 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

4.3.5.5 Constant Registers

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

Register address 0x2000 (MODBUS Address 408193)	
Value	Zero, GP_ZERO
Access	Read
Description	Constant with zeros

Register address 0x2001 (MODBUS Address 408194)	
Value	Ones, GP_ONES
Access	Read
Description	Constant with ones. Is –1 if this is declared as "signed int" or MAXVALUE if it is declared as "unsigned int".

Register address 0x2002 (MODBUS Address 408195)	
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.

Register address 0x2003 (MODBUS Address 408196)	
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.

Register address 0x2004 (MODBUS Address 408197)	
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.

Register address 0x2005 (MODBUS Address 408198)	
Value	Maximum positive number, GP_MAX_POS
Access	Read
Description	Constant in order to control arithmetic.

Register address 0x2006 (MODBUS Address 408199)	
Value	Maximum negative number, GP_MAX_NEG
Access	Read
Description	Constant in order to control arithmetic.

Register address 0x2007 (MODBUS Address 408200)	
Value	Maximum half positive number, GP_HALF_POS
Access	Read
Description	Constant in order to control arithmetic.

Register address 0x2008 (MODBUS Address 408201)	
Value	Maximum half negative number, GP_HALF_NEG
Access	Read
Description	Constant in order to control arithmetic.

Register address 0x3000 to 0x5FFF (MODBUS Address 412289 to 424576)	
Value	Retain range
Access	read/write
Description	These registers can be accessed as the flag/retain range.

5 I/O Modules

5.1 Overview

All listed bus modules, in the overview below, are available for modular applications with the WAGO-I/O-SYSTEM 750.
For detailed information on the I/O modules and the module variations, please refer to the manuals for the I/O modules.
You will find these manuals on CD ROM „ELECTRONICC Tools and Docs“ (Item No.: 0888-0412) or at http://www.wago.com under Documentation.

Additional Information
Current information on the modular WAGO-I/O-SYSTEM is available at http://www.wago.com.

5.1.1 Digital Input Modules

Tab. 5-1: Digital input modules

DI DC 5 V	
750-414	4 Channel, DC 5 V, 0.2 ms, 2- to 3-conductor connection, high-side switching
DI DC 5(12) V	
753-434	8 Channel, DC 5(12) V, 0.2 ms, 1-conductor connection, high-side switching
DI DC 24 V	
750-400, 753-400	2 Channel, DC 24 V, 3.0 ms, 2- to 4-conductor connection; high-side switching
750-401, 753-401	2 Channel, DC 24 V, 0.2 ms, 2- to 4-conductor connection; high-side switching
750-410, 753-410	2 Channel, DC 24 V, 3.0 ms, 2- to 4-conductor connection; high-side switching
750-411, 753-411	2 Channel, DC 24 V, 0.2 ms, 2- to 4-conductor connection; high-side switching
750-418, 753-418	2 Channel, DC 24 V, 3.0 ms, 2- to 3-conductor connection; high-side switching; diagnostics and confirmation
750-419	2 Channel, DC 24 V, 3.0 ms, 2- to 3-conductor connection; high-side switching; diagnostics
750-421, 753-421	2 Channel, DC 24 V, 3.0 ms, 2- to 3-conductor connection; high-side switching; diagnostics
750-402, 753-402	4 Channel, DC 24 V, 3.0 ms, 2- to 3-conductor connection; high-side switching

750-432, 753-432	4 Channel, DC 24 V, 3.0 ms, 2-conductor connection; high-side switching
750-403, 753-403	4 Channel, DC 24 V, 0.2 ms, 2- to 3-conductor connection; high-side switching
750-433, 753-433	4 Channel, DC 24 V, 0.2 ms, 2-conductor connection; high-side switching
750-422, 753-422	4 Channel, DC 24 V, 2- to 3-conductor connection; high-side switching; 10 ms pulse extension
750-408, 753-408	4 Channel, DC 24 V, 3.0 ms, 2- to 3-conductor connection; low-side switching
750-409, 753-409	4 Channel, DC 24 V, 0.2 ms, 2- to 3-conductor connection; low-side switching
750-430, 753-430	8 Channel, DC 24 V, 3.0 ms, 1-conductor connection; high-side switching
750-431, 753-431	8 Channel, DC 24 V, 0.2 ms, 1-conductor connection; high-side switching
750-436	8 Channel, DC 24 V, 3.0 ms, 1-conductor connection; low-side switching
750-437	8 Channel, DC 24 V, 0.2 ms, 1-conductor connection; low-side switching
DI AC/DC 24 V	
750-415, 753-415	4 Channel, AC/DC 24 V, 2-conductor connection
750-423, 753-423	4 Channel, AC/DC 24 V, 2- to 3-conductor connection; with power jumper contacts
DI AC/DC 42 V	
750-428, 753-428	4 Channel, AC/DC 42 V, 2-conductor connection
DI DC 48 V	
750-412, 753-412	2 Channel, DC 48 V, 3.0ms, 2- to 4-conductor connection; high-side switching
DI DC 110 V	
750-427, 753-427	2 Channel, DC 110 V, configurable high-side or low-side switching
DI AC 120 V	
750-406, 753-406	2 Channel, AC 120 V, 2- to 4-conductor connection; high-side switching
DI AC 120(230) V	
753-440	4 Channel, AC 120(230) V, 2-conductor connection; high-side switching
DI AC 230 V	
750-405, 753-405	2 Channel, AC 230 V, 2- to 4-conductor connection; high-side switching

DI NAMUR	
750-435	1 Channel, NAMUR EEx i, proximity switch acc. to DIN EN 50227
750-425, 753-425	2 Channel, NAMUR, proximity switch acc. to DIN EN 50227
750-438	2 Channel, NAMUR EEx i, proximity switch acc. to DIN EN 50227
DI Intruder Detection	
750-424, 753-424	2 Channel, DC 24 V, intruder detection

5.1.2 Digital Output Modules

Tab. 5-2: Digital output modules

DO DC 5 V	
750-519	4 Channel, DC 5 V, 20mA, short-circuit-protected; high-side switching
DO DC 12(14) V	
753-534	8 Channel, DC 12(14) V, 1A, short-circuit-protected; high-side switching
DO DC 24 V	
750-501, 753-501	2 Channel, DC 24 V, 0.5 A, short-circuit-protected; high-side switching
750-502, 753-502	2 Channel, DC 24 V, 2.0 A, short-circuit-protected; high-side switching
750-506, 753-506	2 Channel, DC 24 V, 0.5 A, short-circuit-protected; high-side switching; diagnostics
750-507, 753-507	2 Channel, DC 24 V, 2.0 A, short-circuit-protected; high-side switching; diagnostics; no longer available, replaced by 750-508!
750-508	2 Channel, DC 24 V, 2.0 A, short-circuit-protected; high-side switching; diagnostics; replacement for 750-507
750-535	2 Channel, DC 24 V, EEx i, short-circuit-protected; high-side switching
750-504, 753-504	4 Channel, DC 24 V, 0.5 A, short-circuit-protected; high-side switching
750-531, 753-531	4 Channel, DC 24 V, 0.5 A, short-circuit-protected; high-side switching
750-532	4 Channel, DC 24 V, 0.5 A, short-circuit-protected; high-side switching; diagnostics
750-516, 753-516	4 Channel, DC 24 V, 0.5 A, short-circuit-protected; low-side switching
750-530, 753-530	8 Channel, DC 24 V, 0.5 A, short-circuit-protected; high-side switching
750-537	8 Channel, DC 24 V, 0.5 A, short-circuit-protected; high-side switching; diagnostics
750-536	8 Channel, DC 24 V, 0.5 A, short-circuit-protected; low-side switching
DO AC 120(230) V	
753-540	4 Channel, AC 120(230) V, 0.25 A, short-circuit-protected; high-side switching

DO AC/DC 230 V	
750-509, 753-509	2 Channel solid state relay, AC/DC 230 V, 300 mA
750-522	2 Channel solid state relay, AC/DC 230 V, 500 mA, 3 A (< 30 s)
DO Relay	
750-523	1 Channel, AC 230 V, AC 16 A, potential-free, 1 make contact
750-514, 753-514	2 Channel, AC 125 V , AC 0.5 A , DC 30 V, DC 1 A, potential-free, 2 changeover contacts
750-517, 753-517	2 Channel, AC 230 V, 1 A, potential-free, 2 changeover contacts
750-512, 753-512	2 Channel, AC 230 V, DC 30 V, AC/DC 2 A, non-floating, 2 make contacts
750-513, 753-513	2 Channel, AC 230 V, DC 30 V, AC/DC 2 A, potential-free, 2 make contacts

5.1.3 Analog Input Modules

Tab. 5-3: Analog input modules

AI 0 - 20 mA	
750-452, 753-452	2 Channel, 0 - 20 mA, differential input
750-465, 753-465	2 Channel, 0 - 20 mA, single-ended
750-472, 753-472	2-Channel, 0 - 20 mA, 16 bit, single-ended
750-480	2-Channel, 0 - 20 mA , differential input
750-453, 753-453	4 Channel, 0 - 20 mA, single-ended
AI 4 - 20 mA	
750-454, 753-454	2 Channel, 4 - 20 mA, differential input
750-474, 753-474	2 Channel, 4 - 20 mA, 16 bit, single-ended
750-466, 753-466	2 Channel, 4 - 20 mA, single ended
750-485	2 Channel, 4 - 20 mA, EEx i, single-ended
750-492, 753-492	2 Channel, 4 - 20 mA, isolated differential input
750-455, 753-455	4 Channel, 4 - 20 mA, single-ended
AI 0 - 1 A	
750-475, 753-475	2-Channel, 0 - 1 A AC/DC, differential input
AI 0 - 5 A	
750-475/020-000, 753-475/020-000	2-Channel, 0 - 5 A AC/DC, differential input

AI 0 - 10 V	
750-467, 753-467	2 Channel, DC 0 - 10 V, single-ended
750-477, 753-477	2 Channel, AC/DC 0 - 10 V, differential input
750-478, 753-478	2 Channel, DC 0 - 10 V, single-ended
750-459, 753-459	4 Channel, DC 0 - 10 V, single-ended
750-468	4 Channel, DC 0 - 10 V, single-ended
AI DC ± 10 V	
750-456, 753-456	2 Channel, DC ± 10 V, differential input
750-479, 753-479	2 Channel, DC ± 10 V, differential measurement input
750-476, 753-476	2 Channel, DC ± 10 V, single-ended
750-457, 753-457	4 Channel, DC ± 10 V, single-ended
AI DC 0 - 30 V	
750-483, 753-483	2 Channel, DC 0 -30 V, differential measurement input
AI Resistance Sensors	
750-461, 753-461	2 Channel, resistance sensors, PT100 / RTD
750-481/003-000	2 Channel, resistance sensors, PT100 / RTD, EEx i
750-460	4 Channel, resistance sensors, PT100 / RTD
AI Thermocouples	
750-462	2 Channel, thermocouples, line break detection, sensor types: J, K, B, E, N, R, S, T, U
750-469, 753-469	2 Channel, thermocouples, line break detection, sensor types: J, K, B, E, N, R, S, T, U, L
AI Others	
750-491	1 Channel for resistor bridges (strain gauge)

5.1.4 Analog Output Modules

Tab. 5-4: Analog output modules

AO 0 - 20 mA	
750-552, 753-552	2 Channel, 0 - 20 mA
750-585	2 Channel, 0 - 20 mA, EEx i
750-553, 753-553	4 Channel, 0 - 20 mA
AO 4 - 20 mA	
750-554, 753-554	2 Channel, 4 - 20 mA
750-554, 753-554	4 Channel, 4 - 20 mA
AO DC 0 - 10 V	
750-550, 753-550	2 Channel, DC 0 - 10 V
750-560	2 Channel, DC 0 - 10 V, 10 bit, 100 mW, 24 V
750-559, 753-559	4 Channel, DC 0 - 10 V
AO DC ± 10 V	
750-556, 753-556	2 Channel, DC ± 10 V
750-557, 753-557	4 Channel, DC ± 10 V

5.1.5 Special Modules

Tab. 5-5: Special modules

Counter Modules	
750-404, 753-404	Up / down counter, DC 24 V, 100 kHz
750-638, 753-638	2 Channel, up / down counter, DC 24 V/ 16 bit / 500 Hz
Frequency Measuring	
750-404/000-003, 753-404/000-003	Frequency measuring
Pulse Width Module	
750-511	2-channel pulse width module, DC 24 V, short-circuit-protected, high-side switching
Distance and Angle Measurement Modules	
750-630	SSI transmitter interface
750-631	Incremental encoder interface, differential inputs
750-634	Incremental encoder interface, DC 24 V
750-637	Incremental encoder interface RS 422, cam outputs
750-635, 753-635	Digital pulse interface, for magnetostrictive distance sensors
Serial Interfaces	
750-650, 753	Serial interface RS 232 C
750-653, 753	Serial interface RS 485
750-651	TTY-Serial interface, 20 mA Current Loop
750-654	Data exchange module
DALI / DSI Master Module	
750-641	DALI / DSI master module
AS interface Master Module	
750-655	AS interface master module
Radio Receiver Module	
750-642	Radio receiver EnOcean
MP Bus Master Module	
750-643	MP bus (multi point bus) master module
Vibration Monitoring	
750-645	2 Channel vibration velocity / bearing condition monitoring VIB I/O

PROFIsafe Modules	
750-660/000-001	8FDI 24V DC PROFIsafe; PROFIsafe 8 channel digital input module
750-665/000-001	4FDO 0.5A / 4FDI 24V DC PROFIsafe; PROFIsafe 4 channel digital input and output module
750-666/000-001	1FDO 10A / 2FDO 0.5A / 2FDI 24V PROFIsafe; PROFIsafe power switch module
RTC Module	
750-640	RTC module
KNX / EIB TP1 Module	
750-646	KNX / EIB /TP1 module – device mode / router mode

5.1.6 System Modules

Tab. 5-6: System modules

Module Bus Extension	
750-627	Module bus extension, end module
750-628	Module bus extension, coupler module
DC 24 V Power Supply Modules	
750-602	DC 24 V, passive
750-601	DC 24 V, max. 6.3 A, without diagnostics, with fuse-holder
750-610	DC 24 V, max. 6.3 A, with diagnostics, with fuse-holder
750-625	DC 24 V, EEx i, with fuse-holder
DC 24 V Power Supply Modules with bus power supply	
750-613	Bus power supply, 24 V DC
AC 120 V Power Supply Modules	
750-615	AC 120 V, max. 6.3 A without diagnostics, with fuse-holder
AC 230 V Power Supply Modules	
750-612	AC/DC 230 V without diagnostics, passive
750-609	AC 230 V, max. 6.3 A without diagnostics, with fuse-holder
750-611	AC 230 V, max. 6.3 A with diagnostics, with fuse-holder
Filter Modules	
750-624	Filter module, field side power supply
750-626	Filter module, system and field side power supply
Field Side Connection Module	
750-603, 753-603	Field side connection module, DC 24 V
750-604, 753-604	Field side connection module, DC 0 V
750-614, 753-614	Field side connection module, AC/DC 0 ... 230 V
Separation Modules	
750-616	Separation module
750-621	Separation module with power contacts
Binary Spacer Module	
750-622	Binary spacer module
End Module	
750-600	End module, to loop the internal bus

5.2 Process Data Architecture for MODBUS/TCP

With some I/O modules, the structure of the process data is fieldbus specific.

In the case of a coupler/controller with MODBUS/TCP, the process image uses a word structure (with word alignment). The internal mapping method for data greater than one byte conforms to the Intel format.

The following section describes the process image for various WAGO-I/O-SYSTEM 750 and 753 I/O modules when using a coupler/controller with MODBUS/TCP.

Note
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.

For the PFC process image of the programmable fieldbus controller is the structure of the process data mapping identical.

5.2.1 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.

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.

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).

1 Channel Digital Input Module with Diagnostics

750-435

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

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

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

2 Channel Digital Input Modules with Diagnostics

750-419, -421, -424, -425, 753-421, -424, -425

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

2 Channel Digital Input Module with Diagnostics and Output Process Data

750-418, 753-418

The 750-418, 753-418 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.

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

Output Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				Acknowledgement bit Q 2 Channel 2	Acknowledgement bit Q 1 Channel 1	0	0

4 Channel Digital Input Modules

750-402, -403, -408, -409, -414, -415, -422, -423, -428, -432, -433,
753-402, -403, -408, -409, -415, -422, -423, -428, -432, -433, -440

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

8 Channel Digital Input Modules

750-430, -431, -436, -437, 753-430, -431, -434

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

5.2.2 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.

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.

1 Channel Digital Output Module with Input Process Data

750-523

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

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

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

2 Channel Digital Input Modules with Diagnostics and Input Process Data

750-507 (-508), -522, 753-507

The 750-507 (-508), -522 and 753-507 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.

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 750-506, 753-506 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.

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

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

4 Channel Digital Output Modules

750-504, -516, -519, -531, 753-504, -516, -531, -540

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

4 Channel Digital Output Modules with Diagnostics and Input Process Data

750-532

The 750-532 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.

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 3 Channel 4	Diagnostic bit S 2 Channel 3	Diagnostic bit S 1 Channel 2	Diagnostic bit S 0 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

8 Channel Digital Output Module

750-530, -536, 753-530, -434

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

8 Channel Digital Output Modules with Diagnostics and Input Process Data

750-537

The 750-537 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.

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Diagnostic bit S 7 Channel 8	Diagnostic bit S 6 Channel 7	Diagnostic bit S 5 Channel 6	Diagnostic bit S 4 Channel 5	Diagnostic bit S 3 Channel 4	Diagnostic bit S 2 Channel 3	Diagnostic bit S 1 Channel 2	Diagnostic bit S 0 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

5.2.3 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.

1 Channel Analog Input Module

750-491, (and all variations)

Input Process Image			
Offset	Byte Destination		Remark
	High Byte	Low Byte	
0	D1	D0	Measured Value U_D
1	D3	D2	Measured Value U_{ref}

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)

Input Process Image			
Offset	Byte Destination		Remark
	High Byte	Low Byte	
0	D1	D0	Measured Value Channel 1
1	D3	D2	Measured Value Channel 2

4 Channel Analog Input Modules

750-453, -455, -457, -459, -460, -468, (and all variations),
753-453, -455, -457, -459

Input Process Image			
Offset	Byte Destination		Remark
	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

5.2.4 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.

2 Channel Analog Output Modules

750-550, -552, -554, -556, -560, -585, (and all variations),
753-550, -552, -554, -556

Output Process Image			
Offset	Byte Destination		Remark
	High Byte	Low Byte	
0	D1	D0	Output Value Channel 1
1	D3	D2	Output Value Channel 2

4 Channel Analog Output Modules

750-553, -555, -557, -559, 753-553, -555, -557, -559

Output Process Image			
Offset	Byte Destination		Remark
	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

5.2.5 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.

Further information
For detailed information about the structure of a particular module's control/status byte, please refer to that module's manual. Manuals for each module can be found on the Internet under:
http://www.wago.com.

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.

Input Process Image			
Offset	Byte Destination		Remark
	High Byte	Low Byte	
0	-	S	Status byte
1	D1	D0	Counter Value
2	D3	D2	

Output Process Image			
Offset	Byte Destination		Remark
	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.

Input Process Image			
Offset	Byte Destination		Remark
	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		Remark
	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.

Input Process Image			
Offset	Byte Destination		Remark
	High Byte	Low Byte	
0	-	S0	Status byte of Counter 1
1	D1	D0	Counter Value of Counter 1
2	-	S1	Status byte of Counter 2
3	D3	D2	Counter Value of Counter 2

Output Process Image			
Offset	Byte Destination		Remark
	High Byte	Low Byte	
0	-	C0	Control byte of Counter 1
1	D1	D0	Counter Setting Value of Counter 1
2	-	C1	Control byte of Counter 2
3	D3	D2	Counter Setting Value of Counter 2

Pulse Width Modules

750-511, (and all variations)

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.

Input and Output Process Image			
Offset	Byte Destination		Remark
	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

Serial Interface Modules with alternative Data Format

750-650, (and the variations /000-002, -004, -006, -009, -010, -011, -012, -013)
750-651, (and the variations /000-002, -003)
750-653, (and the variations /000-002, -007)

Note
With the freely parametrizable variations /003 000 of the serial interface modules, the desired operation mode can be set. Dependent on it, the process image of these modules is then the same, as from the appropriate variation.

The above Serial Interface Modules with alternative data format have a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 2 words mapped into each image. Word alignment is applied.

Input and Output Process Image				
Offset	Byte Destination		Remark	
	High Byte	Low Byte		
0	D0	C/S	Data byte	Control/Status byte
1	D2	D1	Data bytes	

Serial Interface Modules with Standard Data Format

750-650/000-001, -014, -015, -016
750-651/000-001
750-653/000-001, -006

The above Serial Interface Modules with Standard Data Format have a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 3 words mapped into each image. Word alignment is applied.

Input and Output Process Image				
Offset	Byte Destination		Remark	
	High Byte	Low Byte		
0	D0	C/S	Data byte	Control/Status byte
1	D2	D1	Data bytes	
2	D4	D3		

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.

Input and Output Process Image			
Offset	Byte Destination		Remark
	High Byte	Low Byte	
0	D1	D0	Data bytes
1	D3	D2	

SSI Transmitter Interface Modules

750-630, (and all variations)

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.

Input Process Image			
Offset	Byte Destination		Remark
	High Byte	Low Byte	
0	D1	D0	Data bytes
1	D3	D2	

Incremental Encoder Interface Modules

750-631

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.

Input Process Image				
Offset	Byte Destination		Remark	
	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		Remark	
	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.

Input Process Image				
Offset	Byte Destination		Remark	
	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		Remark	
	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.

Input and Output Process Image			
Offset	Byte Destination		Remark
	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.

Input and Output Process Image				
Offset	Byte Destination		Remark	
	High Byte	Low Byte		
0	D0	C0/S0	Data byte	Control/Status byte
1	D2	D1	Data bytes	

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.

Input and Output Process Image				
Offset	Byte Destination		Remark	
	High Byte	Low Byte		
0	ID	C/S	Command byte	Control/Status byte
1	D1	D0	Data bytes	
2	D3	D2		

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.

Input Process Image				
Offset	Byte Destination		Remark	
	High Byte	Low Byte		
0	D0	S	DALI Response	Status byte
1	D2	D1	Message 3	DALI Address
3	D4	D3	Message 1	Message 2

Output Process Image				
Offset	Byte Destination		Remark	
	High Byte	Low Byte		
0	D0	C	DALI command, DSI dimming value	Control byte
1	D2	D1	Parameter 2	DALI Address
3	D4	D3	Command-Extension	Parameter 1

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.

Input Process Image				
Offset	Byte Destination		Remark	
	High Byte	Low Byte		
0	D0	S	Data byte	Status byte
1	D2	D1	Data bytes	

Output Process Image				
Offset	Byte Destination		Remark	
	High Byte	Low Byte		
0	-	C	not used	Control byte
1	-	-	not used	

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.

Input and Output Process Image				
Offset	Byte Destination		Remark	
	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		

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.

Input and Output Process Image				
Offset	byte Destination		Remark	
	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 1)	
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)	

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.

<table>
<tr><th colspan="5">Input and Output Process Image</th></tr>
<tr><th rowspan="2">Offset</th><th colspan="2">Byte Destination</th><th colspan="2" rowspan="2">Remark</th></tr>
<tr><th>High Byte</th><th>Low Byte</th></tr>
<tr><td>0</td><td>-</td><td>C0/S0</td><td>not used</td><td>Control/Status byte</td></tr>
<tr><td>1</td><td>D1</td><td>D0</td><td colspan="2" rowspan="5">Mailbox (0, 3, 5, 6 or 9 words) / Process data (0-16 words)</td></tr>
<tr><td>2</td><td>D3</td><td>D2</td></tr>
<tr><td>3</td><td>D5</td><td>D4</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>max. 23</td><td>D45</td><td>D44</td></tr>
</table>

5.2.6 System Modules

System Modules with Diagnostics

750-610, -611

The 750-610 and 750-611 Supply Modules provide 2 bits of diagnostics in the Input Process Image for monitoring of the internal power supply.

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 Voltage

Binary Space Module

750-622

The Binary Space Modules 750-622 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.

Input or 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

6 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.

6.1 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.

6.2 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.

6.3 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 are 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.

6.4 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.

6.5 APDU_Segment_Timeout

Data Type Unsigned

Objects 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.

6.6 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.

6.7 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.

6.8 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.

6.9 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.

6.10 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.

6.11 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.

6.12 Configuration_Files

Date 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.

6.13 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.

6.14 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.

6.15 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.

6.16 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.

6.17 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.

6.18 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.

6.19 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.

6.20 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.

6.21 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.

6.22 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.

6.23 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.

6.24 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.

6.25 Event_Time_Stamps

Data Type BACnetARRAY[3] of 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.

6.26 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.

6.27 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.

6.28 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}.

6.29 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.

6.30 File_Type

Data Type CharacterString

Objects File

Description This property, of type CharacterString, identifies the intended use of this file.

6.31 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.

6.32 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.

6.33 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 are 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.

6.34 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.

6.35 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.

6.36 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.

6.37 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.

6.38 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.

6.39 Location

Data Type	CharacterString
Objects	Device
Description	This property, of type CharacterString, indicates the physical location of the BACnet Device.

6.40 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.

6.41 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.

6.42 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.

6.43 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.

6.44 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.

6.45 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".

6.46 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".

6.47 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.

6.48 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.

6.49 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.

6.50 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.

6.51 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.

6.52 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.

6.53 Object_List

Date 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.

6.54 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.

6.55 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.

6.56 Out_Of_Service

Data Type BOOLEAN

Objects Binary input, Binary output, Analog input, Analog output, Scheduler

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.

6.57 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 Tab. 6-7).

Binary Output:
If Out_Of_Service is TRUE, then the Polarity property has no effect on the physical output state (see Tab. 6-7).

Tab. 6-7: Interrelationships for the Polarity property

Present_Value	Polarity	Physical state of the input or output	Physical state of device
INACTIVE	NORMAL	OFF or INACTIVE	not running
ACTIVE	NORMAL	ON or ACTIVE	running
INACTIVE	REVERSE	ON or ACTIVE	not running
ACTIVE	REVERSE	OFF or INACTIVE	running

6.58 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 Tab. 6-8.

Tab. 6-8: Interrelationships for the Polarity property (Binary Input)

Present_Value	Polarity	Physical state of the input	Physical state of device
INACTIVE	NORMAL	OFF or INACTIVE	Not in operation
ACTIVE	NORMAL	ON or ACTIVE	In operation
INACTIVE	REVERSE	ON or ACTIVE	Not in operation
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 Tab. 6-9.

Tab. 6-9: Interrelationships for BACnet Polarity (Binary Output)

Present_Value	Polarity	Physical state of the output	Physical state of device
INACTIVE	NORMAL	OFF or INACTIVE	Not in operation
ACTIVE	NORMAL	ON or ACTIVE	In operation
INACTIVE	REVERSE	ON or ACTIVE	Not in operation
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 6.56, "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 6.26, "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.

6.59 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 **4.2.1.2.3.2** for a description of the prioritization mechanism.

6.60 Priority_For_Writing

Data Type Unsigned (1...16)

Objects Scheduler

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.

6.61 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.

6.62 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.

6.63 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.

6.64 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.

6.65 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).

6.66 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.

6.67 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.

6.68 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.

6.69 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).

6.70 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 6.58, "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.

6.71 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}

6.72 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).

6.73 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}

6.74 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.

6.75 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.

6.76 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.

6.77 Units

Data Type	BACnetEngineeringUnits
Objects	Analog Input, Analog Output
Description	This property, of the type BACnetEngineeringUnits, indicates the measurement units of this object.

Additional Information
The list of standardized Engineering Units can be found in the documentation for the BACnet Library "BACnet_xx.lib" on the web site http://www.wago.com under Service → Downloads → Building Automation → BACnet Downloads → Software

6.78 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.

6.79 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.

6.80 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.

6.81 Vendor_Name

Data Type CharacterString

Objects Device

Description This property, of the type CharacterString, identifies the manufacturer of the BACnet Device.

6.82 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 BACnetDaily-Schedule. 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.

7 Protocol Implementation Conformance Statement (PICS)

All devices conforming to the BACnet protocol shall have a Protocol Implementation Conformance Statement (PICS) that identifies all of the portions of BACnet that are implemented.

7.1 PICS Content

A PICS is a written document, created by the manufacturer of a device that identifies the particular options specified by BACnet that are implemented in the device. A BACnet PICS is considered a public document that is available for use by any interested party. 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.

Additional Information
The PICS document for the BACnet/IP Controller 750-830 is provided at the website http://www.wago.com Documentation → WAGO-I/O-SYSTEM 750 → Fieldbus couplers and programmable fieldbus controllers → 750-830 → Additional information

8 Application Examples

8.1 Test of MODBUS Protocol and Fieldbus Nodes

A MODBUS master is required to test for proper functioning of fieldbus nodes. Various PC applications from different manufacturers are offered for this purpose, and you can download some of these from the internet as free demo versions.

A program that is very suitable for testing your ETHERNET TCP/IP fieldbus nodes, e.g. **ModScan** by **Win-Tech.**

Additional Information
A free demo version of ModScan32 and other utilities from Win-Tech area available online:
http://www.win-tech.com/html/demos.htm.

ModScan32 is a Windows application that functions as a MODBUS master. Use this program to access data points for linked ETHERNET TCP/IP fieldbus nodes and make any required changes.

Additional information
For a sample description of how to use the software, go to:
http://www.win-tech.com/html/modscan32.htm

8.2 Visualization and Control Using SCADA Software

This section can and should provide only limited insight on using the WAGO Fieldbus Controller with standard application software for process visualization and control.

There is a wide range of process visualization programs (SCADA software) available from a number of manufacturers.

Additional information
To view a wide selection of SCADA products, go to:
www.iainsider.co.uk/scadasites.htm

SCADA is the acronym for "Supervisory Control and Data Acquisition".

This is a user-oriented tool that is used in close touch with production as a production data system for the sectors automation technology, process control and production monitoring.

SCADA Systems functions include: visualization and monitoring, data access, trend recording, event and alarm processing, process analysis and specific intervention in a process (control).

The WAGO ETHERNET fieldbus nodes provide the requisite process input and output values for this.

Attention
When selecting appropriate SCADA software, always ensure that a MODBUS device driver is available and that the MODBUS TC/IP functions implemented in the controller are supported.

Among the many companies that offer visualization programs with MODBUS device drivers are Wonderware, National Instruments, Think&Do or KEPware Inc. Some of these programs are also available free of charge online as demo versions.

This program requires special use.
The sections below therefore provide a number of essential steps that illustrate how an application can be developed using a WAGO ETHERNET fieldbus node and SCADA software.

- The MODBUS driver must first be loaded; then, select MODBUS ETHERNET.

- The user is then requested to enter the IP address for addressing of the fieldbus node.
  Some programs also allow assigning the node an alias for this step; e.g., designating the node as "Measured data." After this, addressing may only be conducted using this name.

- A graphic object can then be created, such as a switch (digital), or a potentiometer (analog).
  This object is displayed on the user interface and the user must then link it to the required data point at the node.

- Linking is performed by inputting the node address (IP address or alias) of the required MODBUS function code (register/bit read/write) and the MODBUS address for the required channel.
  Again, input is made as required by the specific program.
  Depending on the application software being used, the MODBUS addresses for the bus module may be displayed with 3, or as illustrated in the examples below, with 5 places.

Example for MODBUS function codes

The MODBUS function codes for National Instruments' SCADA software "Lookout," employ 6-bit encoding. For example, with the first bit here representing the function code:

Tab. 8-10: Example for MODBUS Function Codes

Input code	MODBUS function code	
0	FC1 ⇔ read coils	Reading several input bits
1	FC2 ⇔ read input discretes	Reading several input bits
3	FC3 ⇔ read multiple registers	Reading several input registers
4	FC4 ⇔ read input registers	Reading a single input register

The subsequent five positions indicate the channel number for the consecutively numbered digital or analog input or output channels.

Examples

- Read the first digital input: e.g., 0 0000 1
- Read the second analog input: e.g., 3 0000 2

Application example

The digital input channel 2 of the "Measured data" node can be read out by inputting "Measured data . 0 0000 2".

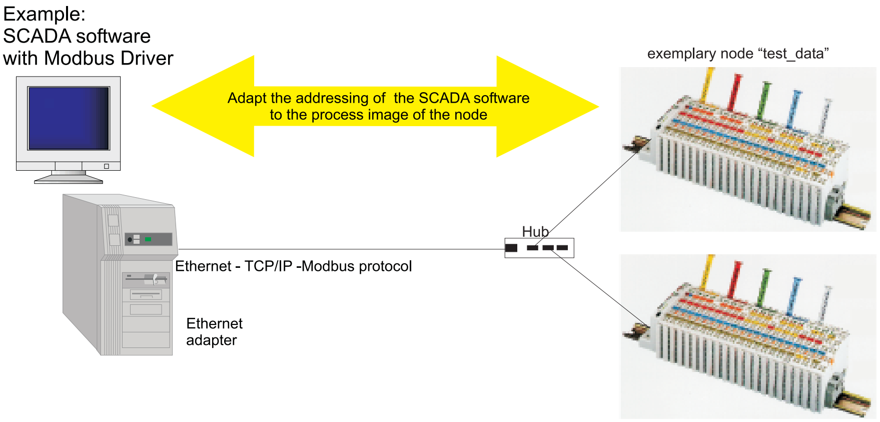


Fig. 8-1: Example of user software G012913e

Additional Information
For a detailed description of how to use a specific software, refer to the manual included with the corresponding SCADA product.

9 Use in Hazardous Environments

9.1 Foreword

Today's development shows that many chemical and petrochemical companies have production plants, production, and process automation machines in operation which use gas-air, vapor-air and dust-air mixtures which can be explosive. For this reason, the electrical components used in such plants and systems must not pose a risk of explosion resulting in injury to persons or damage to property. This is backed by law, directives or regulations on a national and international scale. **WAGO-I/O-SYSTEM 750** (electrical components) is designed for use in zone 2 explosive environments. The following basic explosion protection related terms have been defined.

9.2 Protective Measures

Primarily, explosion protection describes how to prevent the formation of an explosive atmosphere. For instance by avoiding the use of combustible materials, reducing the concentration levels, ventilation measures, to name but a few. But there are a large number of applications, which do not allow the implementation of primary protection measures. In such cases, the secondary explosion protection comes into play. Following is a detailed description of such secondary measures.

9.3 Classification Meeting CENELEC and IEC

The specifications outlined here are valid for use in Europe and are based on the following standards: EN60079... of CENELEC (European Committee for Electrotechnical Standardization). On an international scale, these are reflected by the IEC 60079-... standards of the IEC (International Electrotechnical Commission).

9.3.1 Divisions

Explosive environments are areas in which the atmosphere can potentially become explosive. The term explosive means a special mixture of ignitable substances existing in the form of air-borne gases, fumes, mist or dust under atmospheric conditions which, when heated beyond a tolerable temperature or subjected to an electric arc or sparks, can produce explosions. Explosive zones have been created to describe the concentrations level of an explosive atmosphere. This division, based on the probability of an explosion occurring, is of great importance both for technical safety and feasibility reasons. Knowing that the demands placed on electrical components permanently employed in an explosive environment have to be much more stringent than those placed on electrical components that are only rarely and, if at all, for short periods, subject to a dangerous explosive environment.

Explosive areas resulting from gases, fumes or mist:

- Zone 0 areas are subject to an explosive atmosphere (> 1000 h /year) continuously or for extended periods.
- Zone 1 areas can expect the occasional occurrence of an explosive atmosphere (> 10 h ≤ 1000 h /year).
- Zone 2 areas can expect the rare or short-term occurrence of an explosive atmosphere (> 0 h ≤ 10 h /year).

Explosive areas subject to air-borne dust:

- Zone 20 areas are subject to an explosive atmosphere (> 1000 h /year) continuously or for extended periods.
- Zone 21 areas can expect the occasional occurrence of an explosive atmosphere (> 10 h ≤ 1000 h /year).
- Zone 22 areas can expect the rare or short-term occurrence of an explosive atmosphere (> 0 h ≤ 10 h /year).

9.3.2 Device Group

In addition, the electrical components for explosive areas are subdivided into two device groups:

Device Group I: Device group I includes electrical components for use in below-ground mining operations as well as above-ground systems potentially endangered by mine gas and/or combustible dust.

Device Group II: Device Group II includes electrical components for use in all other explosive environments. This device group is further subdivided by pertinent combustible gases in the environment.

Subdivision IIA, IIB and IIC takes into account that different materials/substances/gases have various ignition energy characteristic values. For this reason the three sub-groups are assigned representative types of gases:

- IIA – Propane
- IIB – Ethylene
- IIC – Hydrogen

Tab. 9-1: Minimal ignition energy of representative types of gases

Minimal Ignition Energy of Representative Types of Gases				
Explosion group	I	IIA	IIB	IIC
Gases	Methane	Propane	Ethylene	Hydrogen
Ignition energy (µJ)	280	250	82	16

Hydrogen being commonly encountered in chemical plants, frequently the explosion device group IIC is requested for maximum safety.

9.3.3 Unit Categories

A further subdivision for device groups is made into categories according to their area of use (zones).

Tab. 9-2: Unit categories

Device group	Unit category	Equipment or Area of use
I	M1	Fire-damp protection
I	M2	Fire-damp protection
II	1G	Zone 0 equipment (Explosive environment by gas, fumes or mist)
II	2G	Zone 1 equipment (Explosive environment by gas, fumes or mist)
II	3G	Zone 2 equipment (Explosive environment by gas, fumes or mist)
II	1D	Zone 20 equipment (Explosive environment by dust)
II	2D	Zone 21 equipment (Explosive environment by dust)
II	3D	Zone 22 equipment (Explosive environment by dust)

9.3.4 Temperature Classes

The maximum surface temperature for electrical components of device group I is 150 °C (danger due to coal dust deposits) or 450 °C (if there is no danger of coal dust deposit).

In line with the maximum permissible surface temperature for all ignition protection types, the electrical components are subdivided into temperature classes, as far as electrical components of device group II are concerned.

The maximum surface temperature of any electrical equipment must always be less than the ignition temperature of the explosive atmosphere occurring in its area of use.

It goes without saying that an equipment fulfilling the requirements of temperature class T3 is also suitable for use in explosive atmospheres of temperature classes T1 and T2.

Tab. 9-3: Temperature classes

Temperature Classes	**Maximum permissible surface temperature of the equipment**	**Ignition Temperature of the Combustible Materials**
T1	450 °C	> 450 °C
T2	300 °C	> 300 °C to 450 °C
T3	200 °C	> 200 °C to 300 °C
T4	135 °C	> 135 °C to 200 °C
T5	100 °C	>100 °C to 135 °C
T6	85 °C	> 85 °C to 100 °C

The following table represents the division and attributes of the materials to the temperature classes and material groups in percent:

Tab. 9-4: Material groups in percent

<table>
<tr><th colspan="7">Temperature classes</th></tr>
<tr><td>T1</td><td>T2</td><td>T3</td><td>T4</td><td>T5</td><td>T6</td><td>Total*</td></tr>
<tr><td>26.6 %</td><td>42.8 %</td><td>25.5 %</td><td rowspan="2">4.9 %</td><td rowspan="2">0 %</td><td rowspan="2">0.2 %</td><td rowspan="2">432</td></tr>
<tr><td colspan="3">94.9 %</td></tr>
<tr><th colspan="7">Explosion group</th></tr>
<tr><td>IIA</td><td>IIB</td><td>IIC</td><td></td><td></td><td></td><td>Total*</td></tr>
<tr><td>85.2 %</td><td>13.8 %</td><td>1.0 %</td><td></td><td></td><td></td><td>501</td></tr>
</table>

* Number of classified materials

9.3.5 Types of Ignition Protection

Ignition protection defines the special measures to be taken for electrical components in order to prevent the ignition of surrounding explosive atmospheres. For this reason a differentiation is made between the following types of ignition protection:

Tab. 9-5: Types of Ignition Protection

Identifi-cation	CENELEC standard	IEC standard	Explanation	Application
Ex o	EN 60079-6	IEC 60079-6	Oil encapsulation	Zone 1 + 2
Ex p	EN 60079-2	IEC 60079-2	Overpressure encapsulation	Zone 1 + 2
Ex q	EN 60079-5	IEC 60079-5	Sand encapsulation	Zone 1 + 2
Ex d	EN 60079-1	IEC 60079-1	Pressure resistant encapsulation	Zone 1 + 2
Ex e	EN 60079-7	IEC 60079-7	Increased safety	Zone 1 + 2
Ex m	EN 60079-18	IEC 60079-18	Cast encapsulation	Zone 1 + 2
Ex i	EN 60079-11 (Device) EN 60079-25 (System)	IEC 60079-11	Intrinsic safety	Zone 0 + 1 + 2
Ex n	EN 60079-15	IEC 60079-15	Electrical components for zone 2 (see below)	Zone 2

Ignition protection “n" describes exclusively the use of explosion protected electrical components in zone 2. This zone encompasses areas where explosive atmospheres can only be expected to occur rarely or short-term. It represents the transition between the area of zone 1, which requires an explosion protection and safe area in which for instance welding is allowed at any time.

Regulations covering these electrical components are being prepared on a world-wide scale. Organizations such as the PTB in Germany certify that the devices meet the requirements of the European standard EN 60079-15.

Type “n” ignition protection additionally requires electrical components to be marked with the following extended identification:

- **Non-sparking apparatus "nA"**
  Apparatus, which is designed so that the risk of occurrence of arcs or sparks, which can give rise to an ignition hazard during normal operation, is minimized.

- **Apparatus with protected contacts "nC"**
  Apparatus with contacts, which close and open a possibly ignitable circuit, in which the contact mechanism is designed so that ignition of a defined potentially explosive atmosphere is prevented.

- **Energy-limited apparatus "nL"**
  Electrical apparatus, in which circuits and components are designed in accordance with the concept of energy limitation.

Additional Information
For more detailed information please refer to the national and/or international standards, directives and regulations!

9.4 Classifications Meeting the NEC 500

The following classifications as defined in article 500 and 505 of the National Electric Code (NEC) are valid for North America.

9.4.1 Divisions

The "Divisions" describe the degree of probability of whatever type of dangerous situation occurring. Here the following assignments apply:

Tab. 9-6: Divisions

Explosion endangered areas due to combustible gases, fumes, mist and dust:	
Division 1	Encompasses areas in which explosive atmospheres are to be expected occasionally (> 10 h ≤ 1000 h /year) as well as continuously and long-term (> 1000 h /year).
Division 2	Encompasses areas in which explosive atmospheres can be expected rarely and short-term (>0 h ≤ 10 h /year).

9.4.2 Explosion Protection Groups

Electrical components for explosion endangered areas are subdivided in three danger categories:

Tab. 9-7: Explosion Protection Groups

Danger category	**Sub-group**
Class I (gases and fumes)	Group A (Acetylene) Group B (Hydrogen) Group C (Ethylene) Group D (Methane)
Class II (dust)	Group E (Metal dust) Group F (Coal dust) Group G (Flour, starch and cereal dust)
Class III (fibers)	No sub-groups

9.4.3 Temperature Classes

Electrical components for explosive areas are differentiated by temperature classes:

Tab. 9-8: Temperature Classes

Temperature classes	Maximum permissible surface temperature of the equipment	Ignition temperature of the combustible materials
T1	450 °C	> 450 °C
T2	300 °C	> 300 °C to 450 °C
T2A	280 °C	> 280 °C to 300 °C
T2B	260 °C	> 260 °C to 280 °C
T2C	230 °C	>230 °C to 260 °C
T2D	215 °C	>215 °C to 230 °C
T3	200 °C	>200 °C to 215 °C
T3A	180 °C	>180 °C to 200 °C
T3B	165 °C	>165 °C to 180 °C
T3C	160 °C	>160 °C to 165 °C
T4	135 °C	>135 °C to 160 °C
T4A	120 °C	>120 °C to 135 °C
T5	100 °C	>100 °C to 120 °C
T6	85 °C	> 85 °C to 100 °C

9.5 Identification

9.5.1 For Europe

According to CENELEC and IEC

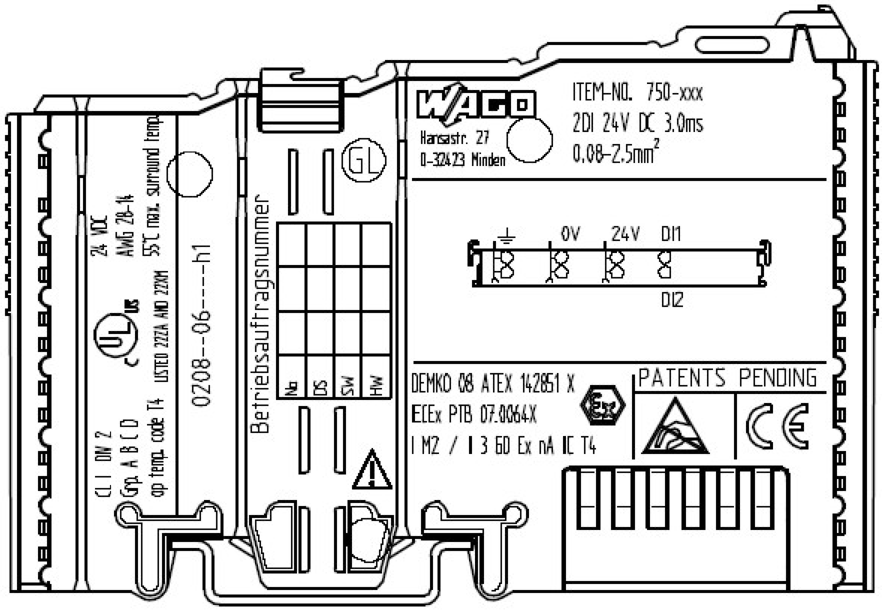


Fig. 9.5.1-1: Example for lateral labeling of bus modules (750-400, 2 channel digital input module 24 V DC) p01xx03x

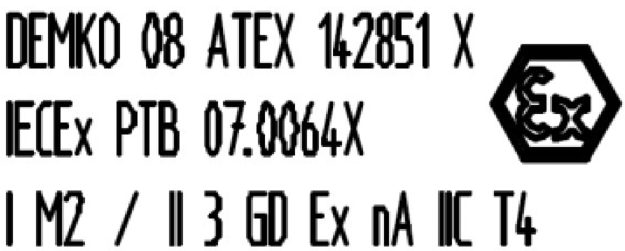


Fig. 9.5.1-2: Printing on text detail in accordance with CENELEC and IEC p01xx04x

Tab. 9-9: Description of Printing on

Printing on Text	Description
DEMKO 08 ATEX 142851 X IECEx PTB 07.0064X	Approval body and/or number of the examination certificate
I M2 / II 3 GD	Device group and Unit category
Ex nA	Type of ignition and extended identification
IIC	Device group
T4	Temperature class

9.5.2 For America

According to NEC 500

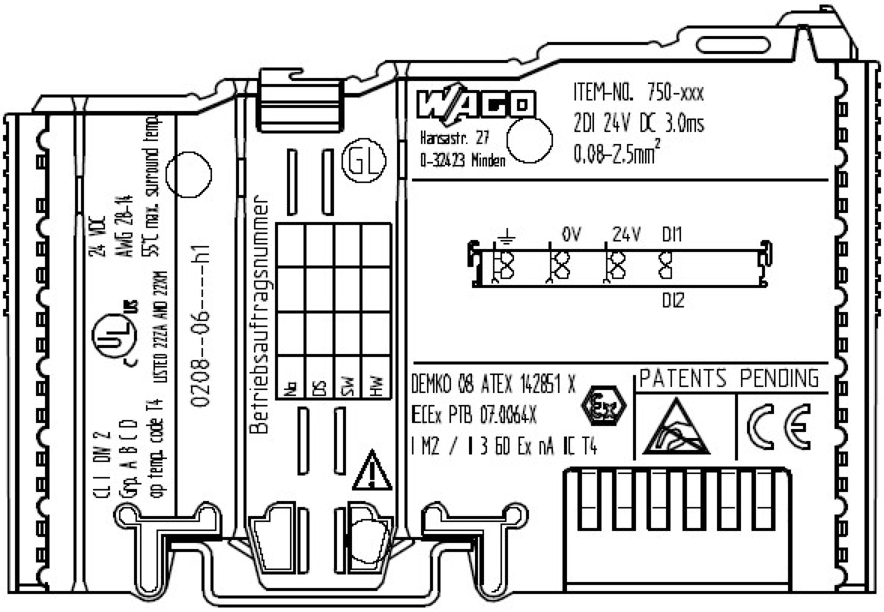


Fig. 9.5.2-3: Example for lateral labeling of bus modules (750-400, 2 channel digital input module 24 V DC) p01xx03x

Fig. 9.5.2-4: Printing on text detail in accordance with CENELEC and IEC p01xx05x

Tab. 9-10: Description of Printing on

Printing on Text	Description
CL 1	Explosion protection group (condition of use category)
DIV 2	Area of application (zone)
Grp. ABCD	Explosion group (gas group)
Op temp. code T4	Temperature class

9.6 Installation Regulations

In the **Federal Republic of Germany,** various national regulations for the installation in explosive areas must be taken into consideration. The basis for this forms the working reliability regulation, which is the national conversion of the European guideline 99/92/E6. They complemented by the installation regulation EN 60079-14. The following are excerpts from additional VDE regulations:

Tab. 9-11: VDE Installation Regulations in Germany

Standard	Installation Regulations
DIN VDE 0100	Installation in power plants with rated voltages up to 1000 V
DIN VDE 0101	Installation in power plants with rated voltages above 1 kV
DIN VDE 0800	Installation and operation in telecommunication plants including information processing equipment
DIN VDE 0185	lightning protection systems

The **USA** and **Canada** have their own regulations. The following are excerpts from these regulations:

Tab. 9-12: Installation Regulations in USA and Canada

Standard	Installation Regulations
NFPA 70	National Electrical Code Art. 500 Hazardous Locations
ANSI/ISA-RP 12.6-1987	Recommended Practice
C22.1	Canadian Electrical Code

Warning
When using the **WAGO-I/O SYSTEM 750** (electrical operation) with Ex approval, the following points are mandatory:

9.6.1 ANSI/ISA 12.12.01

This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only.

Warning
Explosion hazard - substitution of components may impair suitability for Class I, Div. 2.

Warning
Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.

When a fuse is provided, the following marking shall be provided:
”A switch suitable for the location where the equipment is installed shall be provided to remove the power from the fuse.”
The switch need not be integrated in the equipment.

For devices with Ethernet connectors:
”Only for use in LAN, not for connection to telecommunication circuits.”

Warning
Use Module 750-642 only with antenna module 758-910.

9.6.2 TÜV Nord Ex-i applications

For operation in zone 2, the **WAGO-I/O-System 750**-*** must be mounted in an enclosure that fulfills the requirements of the directive 94/9/EG and the relevant standards EN 60079-0 and EN 60079-15. The fulfillment of these requirements must be certified by an appointed office.

Only devices that are suitable for operation in areas subject to explosion of zone 2 and 22 and that are suitable for the conditions in the place of installation (declaration of conformity or certificate from a test office) may be connected to not intrinsically safe circuits in zone 2 or zone 22.

If the interface circuits are operated without the 750-3 fieldbus coupler ../....-... (DEMKO 08 ATEX 142851 X), then measures must be taken outside of the device so that the rated voltage will not be exceeded by more than 40 % due to temporary faults.

The connection and interruption of non intrinsically-safe circuits under voltage is only permissible during installation, maintenance or for repair purposes. The simultaneous occurrence of explosive atmosphere and installation, maintenance, etc. must be ruled out.

For operation in zone 22, the **WAGO-I/O-System 750**-*** must be mounted in an enclosure that fulfills the requirements of the directive 94/9/EG and the relevant standards EN 61241-0 and EN 61241-1. The fulfillment of these requirements must be certified by an appointed office. The housing must be marked on the outside with

II 3 (1) GD Ex nA tD [ia] [iaD] IIC/IIB A22 IP6X T135°C (T4) and/ or

II 3 (2) GD Ex nA tD [ib] [ibD] IIC/IIB A22 IP6X T135°C (T4).

The marking must take into account all devices built into the enclosure.

The manufacturer of the whole device must ensure that taking into account the maximum ambient temperature range, the temperature in the enclosure will not drop below 0 °C or exceed 55 °C.

DIP switches, coding switches, and potentiometers that are connected to the module may only be operated if an explosive atmosphere can be ruled out.

9.6.3 ATEX and IEC Ex

GROUP I, CATEGORY M2 only with a suitable enclosure according to IEC 60079-0 and IEC 60079-1 required by end-user. When used in Category M2 locations, the modules have to be installed in suitable ATEX Category M2 certified enclosures according to EN 60079-0: 2006 and EN 60079-1: 2007.

The Fieldbus Independent Modules of the **WAGO-I/O-System 750**-…/….-…. have to be installed in a Pollution Degree 2 environment or better in the end use application for use with an IP54 minimum enclosure. Except otherwise specified below. Otherwise the modules have to be installed in an IP64 minimum enclosure.

Modules 750-609 and 750-611 have to be installed in an IP 64 minimum enclosure.

When used in the presence of combustible dust the enclosure shall comply with the relevant requirements of IEC 61241-0:2004 and IEC 61241-1:2004.

Installation, addition, removal or replacement of modules, fieldbus connectors or fuses may only take place when the system supply and the field supply are switched off, or when the area is known to be non-hazardous.

DIP-switches, binary-switches and potentiometers attached to the modules may only be adjusted when the area is known to be non-hazardous.

Module 750-642 has to be used only with antenna module 758-910 with a max. cable length of 2.5 m.

Provide the transient protection device not exceeding 40 % of the rated voltage at the power supply terminal of the apparatus.

Ambient temperature range: -20 °C to +55 °C

Additional Information
Proof of certification is available on request.
Also take note of the information given on the module technical information sheet.

The Instruction Manual, containing these special conditions for safe use, must be readily available to the user.

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 a separate, dedicated software and should not be confused with the I/O Configurator for the WAGO I/O-PRO CAA.

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.
Network participants must be registered at the associated BBMD.

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 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 are 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 CAA 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 CAA 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 CAA 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 a 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).

RJ45 Connector

The RJ45 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.

RS232

The RS232 (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 transmission of the data may be randomly long.
As the RS232 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, 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 CAA 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 CAA (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 CAA software is the WAGO I/O PRO 32, Versions 2.1 and 2.2.
The new WAGO-I/O-PRO CAA 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

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
Wiesemann & Theis GmbH
http://www.wut.de

BIG EU (BACnet Interest Group Europe) Last updated March 2007
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 March 2007
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 March 2007
The "Standing Standard Project Committee 135" steers the ongoing development of the protocol
http://www.bacnet.org

Index

F

G

H

I

K

L

M

N

O

P

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

Web: http://www.wago.com