WAGO-I/O-SYSTEM 750

Modular I/O-System

ETHERNET TCP/IP

750-841

Manual

Technical description,
installation and
configuration

Version 1.2.1

WAGO Kontakttechnik GmbH & Co. KG
Hansastraße 27
D-32423 Minden

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

E-Mail: info@wago.com

Web: http://www.wago.com

Technical Support
Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the correctness and completeness of this documentation. However, as errors can never be fully excluded we would appreciate any information or ideas at any time.

E-Mail: documentation@wago.com

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

TABLE OF CONTENTS

1 Important 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 fieldbus 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.

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 ', Nibble separated with dots

1.7 Scope

This manual describes the field bus independent WAGO-I/O-SYSTEM 750 with the programmable fieldbus controller for ETHERNET 10/100 MBit/s.

Item.-No.	Description
750-841	Prog. Fieldbus Controller EtherNet 10/100 MBit/s

1.8 Important Comments for Starting up

Attention
For the start-up of the controller 750-841 important notes are to be considered, because it strongly differentiates in some points of starting up the WAGO ETHERNET controller 750-842.
Read for this the chapter: „ Starting up an ETHERNET TCP/IP fieldbus node“.

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

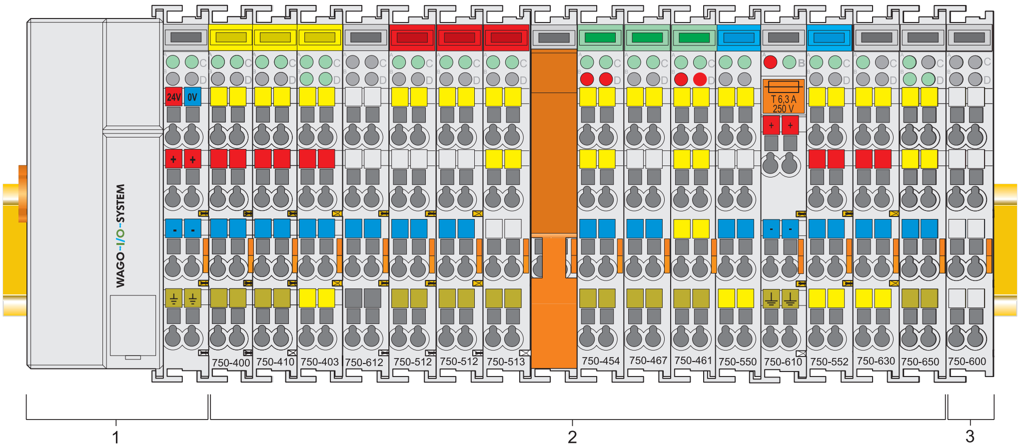


Fig. 2-1: Fieldbus node g0xxx00x

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

The coupler/controller contains the fieldbus interface, electronics and a power supply terminal. The fieldbus interface forms the physical interface to the relevant fieldbus. The electronics process the data of the bus modules and make it available for the fieldbus communication. The 24 V system supply and the 24 V field supply are fed in via the integrated power supply terminal. The fieldbus coupler communicates via the relevant fieldbus. The programmable fieldbus controller (PFC) enables the implementation of additional PLC functions. Programming is done with the WAGO-I/O-PRO 32 in accordance with IEC 61131-3.

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

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

2.2 Technical Data

Mechanic	
Material	Polycarbonate, Polyamide 6.6
Dimensions W x H* x L * from upper edge of DIN 35 rail - Coupler/Controller (Standard) - 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 featherkey-dovetail
Mounting position	any position
Marking	marking label type 247 and 248 paper marking label 8 x 47 mm
Connection	
Connection type	CAGE CLAMP®
Wire range	0.08 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 % to 95 % without condensation
Resistance to harmful substances	acc. to IEC 60068-2-42 and IEC 60068-2-43
Maximum pollutant concentration at relative humidity < 75%	$SO_2 \leq 25$ 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 gasses – ionization radiation.

Safe electrical isolation	
Air and creepage distance	acc. to IEC 60664-1
Degree of pollution acc. To IEC 61131-2	2
Degree of protection	
Degree of protection	IP 20

Electromagnetic compatibility

Immunity to interference for industrial areas acc. to EN 61000-6-2 (2001)

Test specification	Test values		Strength class	Evaluation criteria
EN 61000-4-2 ESD	4 kV/8 kV (contact/air)		2/3	B
EN 61000-4-3 electromagnetic fields	10 V/m 80 MHz ... 1 GHz		3	A
EN 61000-4-4 burst	1 kV/2 kV (data/supply)		2/3	B
EN 61000-4-5 surge	Data:	-/- (line/line)		B
		1 kV (line/earth)	2	
	DC supply:	0.5 kV (line/line)	1	B
		0.5 kV (line/earth)	1	
	AC supply:	1 kV (line/line)	2	B
		2 kV (line/earth)	3	
EN 61000-4-6 RF disturbances	10 V/m 80 % AM (0.15 ... 80 MHz)		3	A

Emission of interference for industrial areas acc. to EN 61000-6-4 (2001)

Test specification	Limit values/[QP]*)	Frequency range	Distance
EN 55011 (AC supply, conducted)	79 dB (µV)	150 kHz ... 500 kHz	
	73 dB (µV)	500 kHz ... 30 MHz	
EN 55011 (radiated)	40 dB (µV/m)	30 MHz ... 230 MHz	10 m
	47 dB (µV/m)	230 MHz ... 1 GHz	10 m

Emission of interference for residential areas acc. to EN 61000-6-3 (2001)

Test specification	Limit values/[QP]*)	Frequency range	Distance
EN 55022 (AC supply, conducted)	66 ... 56 dB (µV)	150 kHz ... 500 kHz	
	56 dB (µV)	500 kHz ... 5 MHz	
	60 dB (µV)	5 MHz ... 30 MHz	
EN 55022 (DC supply/data, conducted)	40 ... 30 dB (µA)	150 kHz ... 500 kHz	
	30 dB (µA)	500 kHz ... 30 MHz	
EN 55022 (radiated)	30 dB (µV/m)	30 MHz ... 230 MHz	10 m
	37 dB (µV/m)	230 MHz ... 1 GHz	10 m

Mechanical strength acc. to IEC 61131-2		
Test specification	**Frequency range**	**Limit value**
IEC 60068-2-6 vibration	5 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 disturances	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
außer für:	24 dB (µV/m)	156 MHz ... 165 MHz	3 m

Mechanical strength acc. to Germanischer Lloyd (2003)

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

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

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

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

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

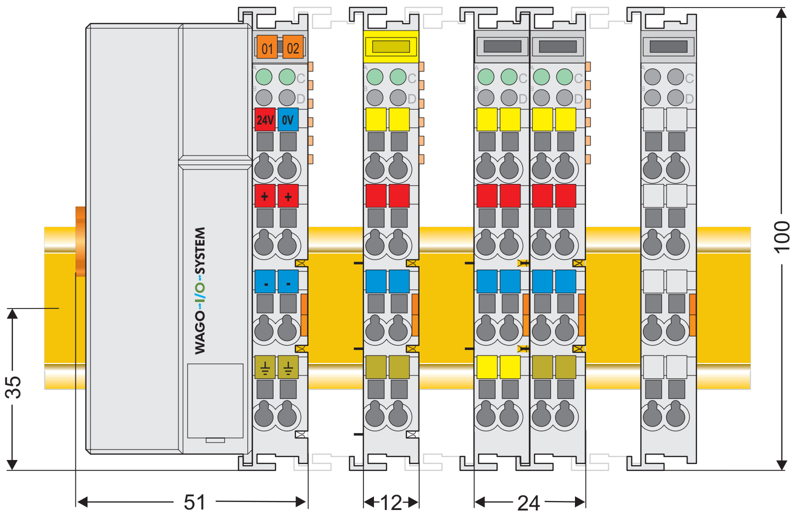
It is possible to use other field bus couplers/controllers under certain boundary conditions. Please contact WAGO Kontakttechnik GmbH & Co. KG.

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

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

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

Dimensions



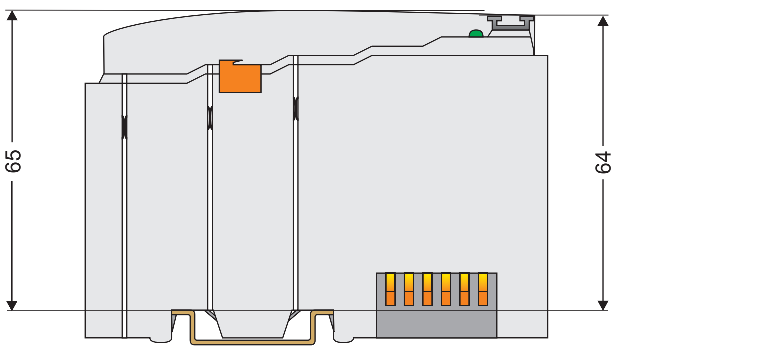


Fig. 2-2: Dimensions g01xx05e

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

2.3 Manufacturing Number

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

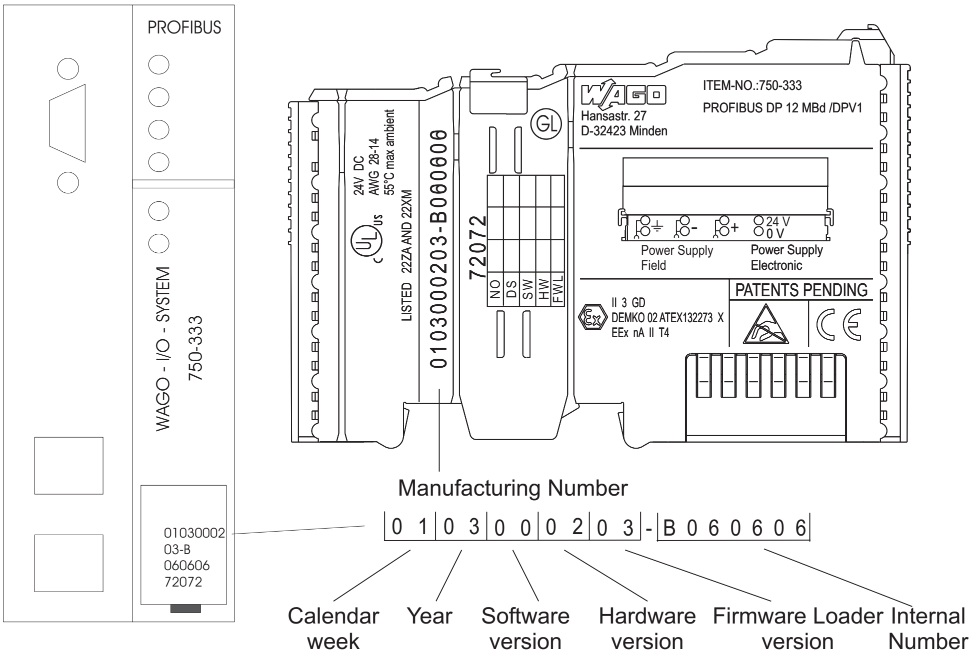


Fig. 2-3: Example: Manufacturing Number of a PROFIBUS fieldbus coupler 750-333

g01xx15e

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

2.4 Component Update

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

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

Update Matrix

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

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

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

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

2.5 Storage, Assembly and Transport

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

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

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

2.6 Mechanical Setup

2.6.1 Installation Position

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

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

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

2.6.2 Total Expansion

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

Examples:

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

Exception:

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

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

2.6.3 Assembly onto Carrier Rail

2.6.3.1 Carrier rail properties

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

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

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

- The material must be non-corrosive.
- Most components have a contact to the carrier rail to ground 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.

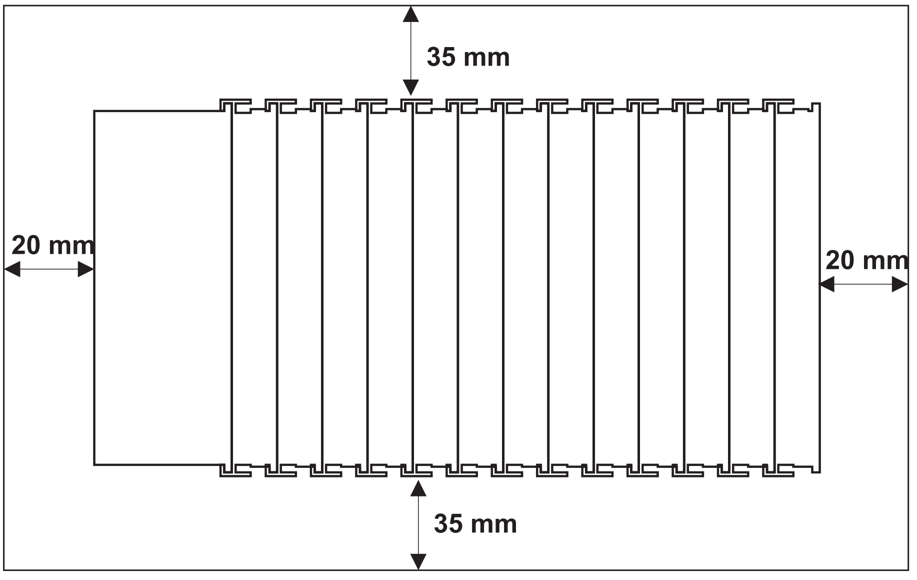


Fig. 2-4: Spacing g01xx13x

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

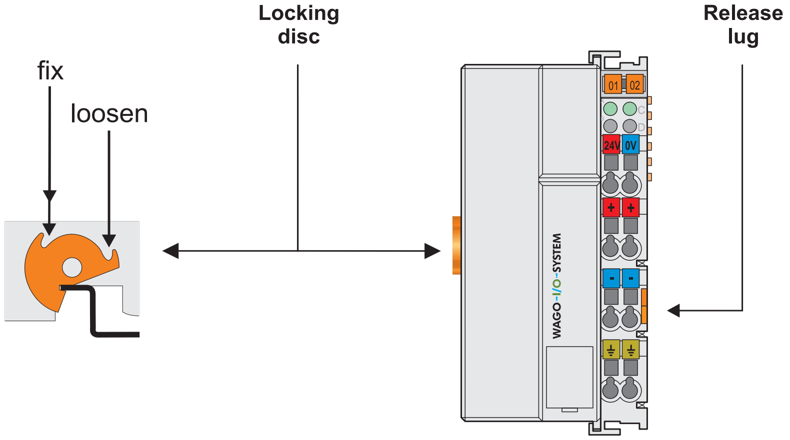


Fig. 2-5: Coupler/Controller and unlocking lug g01xx12e

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

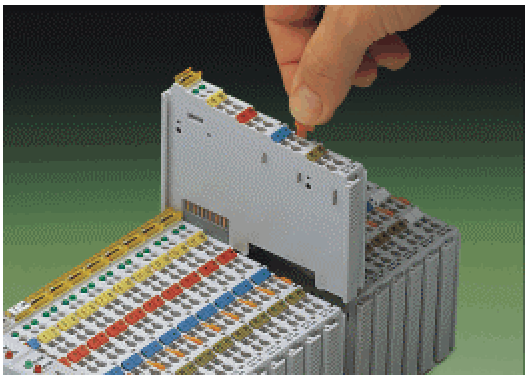


Fig. 2-6: removing bus terminal p0xxx01x

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

2.6.6 Assembly Sequence

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

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

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

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

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

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

2.6.7 Internal Bus/Data Contacts

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

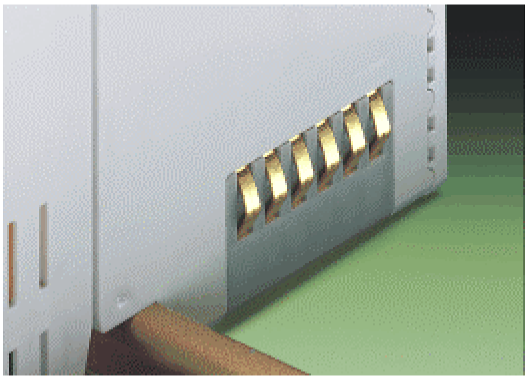


Fig. 2-7: Data contacts p0xxx07x

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

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

2.6.8 Power Contacts

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

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

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

Power jumper contacts

Blade	0	0	3	2
Spring	0	3	3	2

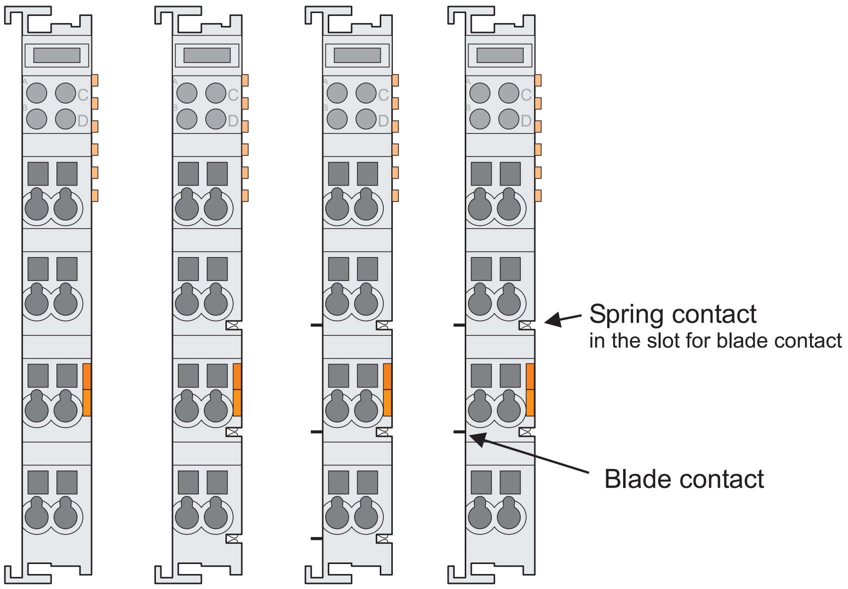


Fig. 2-8: Example for the arrangement of power contacts g0xxx05e

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

2.6.9 Wire connection

All components have CAGE CLAMP® connections.

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

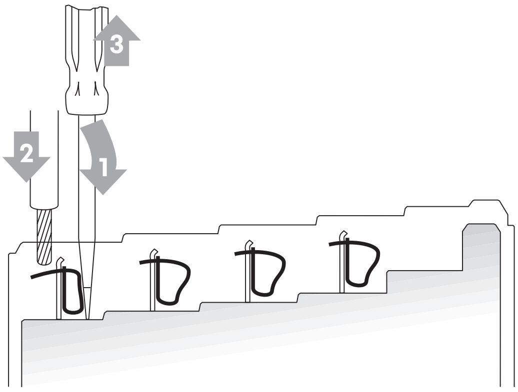


Fig. 2-9: CAGE CLAMP® Connection g0xxx08x

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

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

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

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

2.7 Power Supply

2.7.1 Isolation

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

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

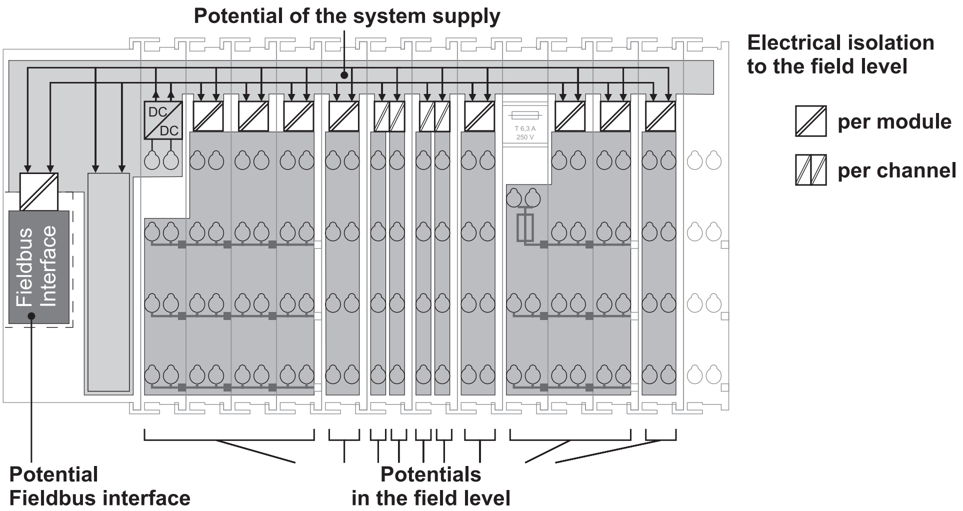


Fig. 2-10: Isolation g0xxx01e

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.

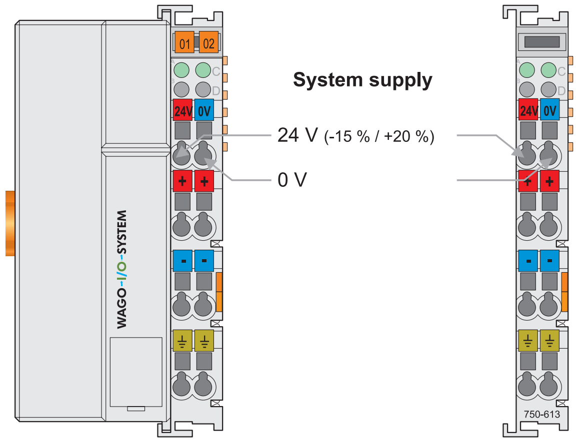


Fig. 2-11: System Supply g0xxx02e

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

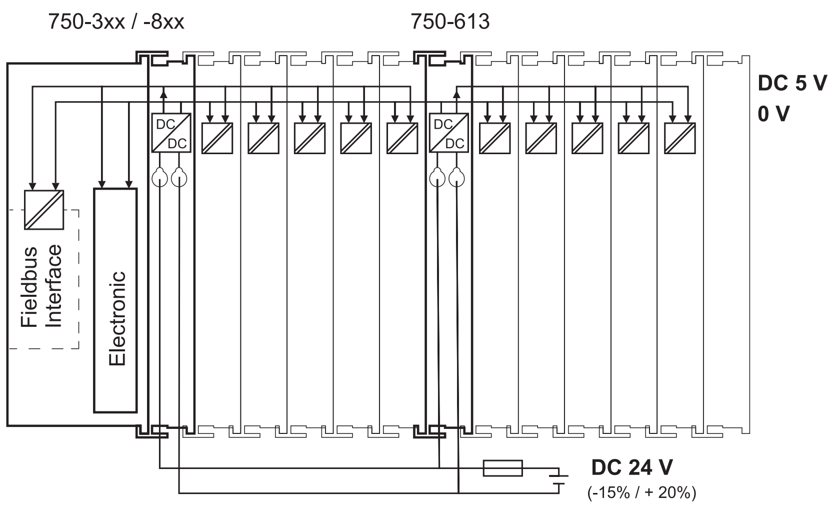


Fig. 2-12: System Voltage g0xxx06e

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

2.7.2.2 Alignment

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

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

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

*) cf. catalogue W4 Volume 3, manuals or Internet

Example

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

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

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

Example:

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

Current consumption:
20* 90 mA = 1800 mA
10* 2 mA = 20 mA
Sum 1820 mA

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

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

The maximum input current of the 24 V system supply is 500 mA. The exact electrical consumption ($I_{(24\ V)}$) 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)

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

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

2.7.3 Field Supply

2.7.3.1 Connection

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

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

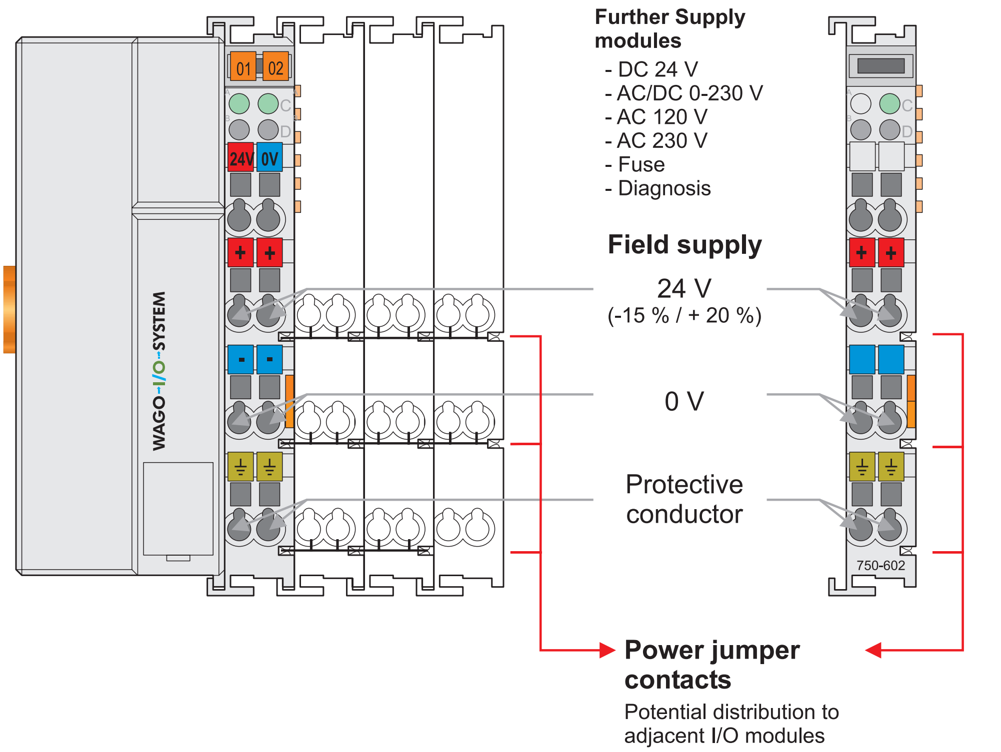


Fig. 2-13: Field Supply (Sensor/Actuator) g0xxx03e

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

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

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

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

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

2.7.3.2 Fusing

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

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

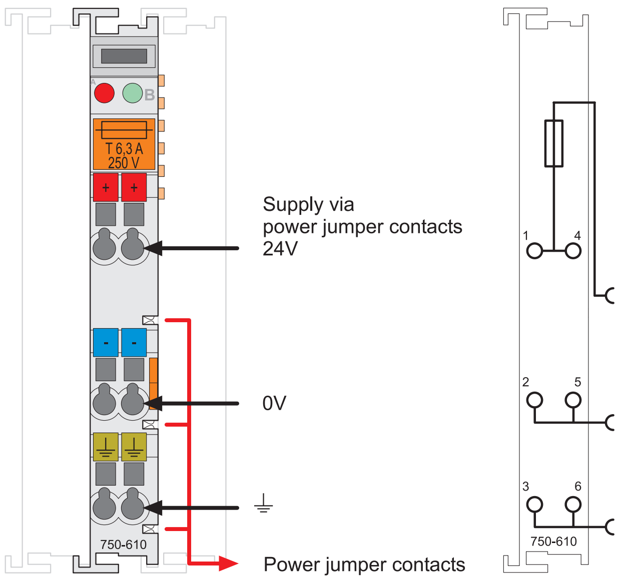


Fig. 2-14: Supply module with fuse carrier (Example 750-610) g0xxx09x

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

For UL approved systems only use UL approved fuses.

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

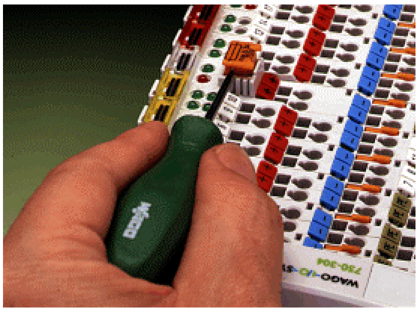


Fig. 2-15: Removing the fuse carrier p0xxx05x

Lifting the cover to the side opens the fuse carrier.

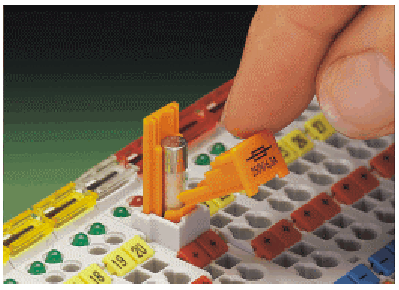


Fig. 2-16: Opening the fuse carrier p0xxx03x

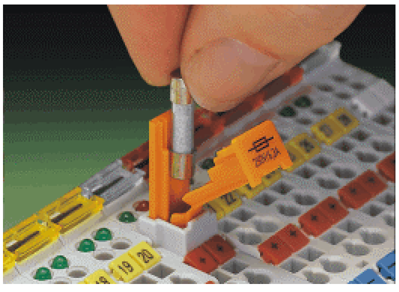


Fig. 2-17: Change fuse p0xxx04x

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

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

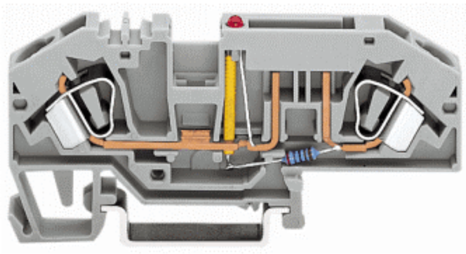


Fig. 2-18: Fuse modules for automotive fuses, Series 282 pf66800x

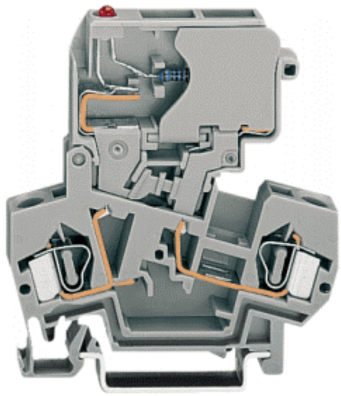


Fig. 2-19: Fuse modules with pivotable fuse carrier, Series 281 pe61100x

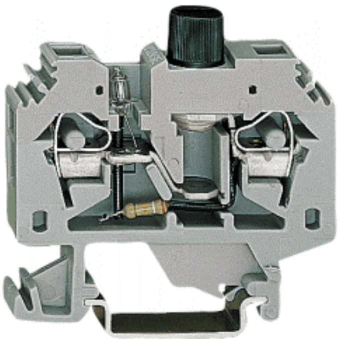


Fig. 2-20: Fuse modules, Series 282 pf12400x

2.7.4 Supplementary power supply regulations

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

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

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

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

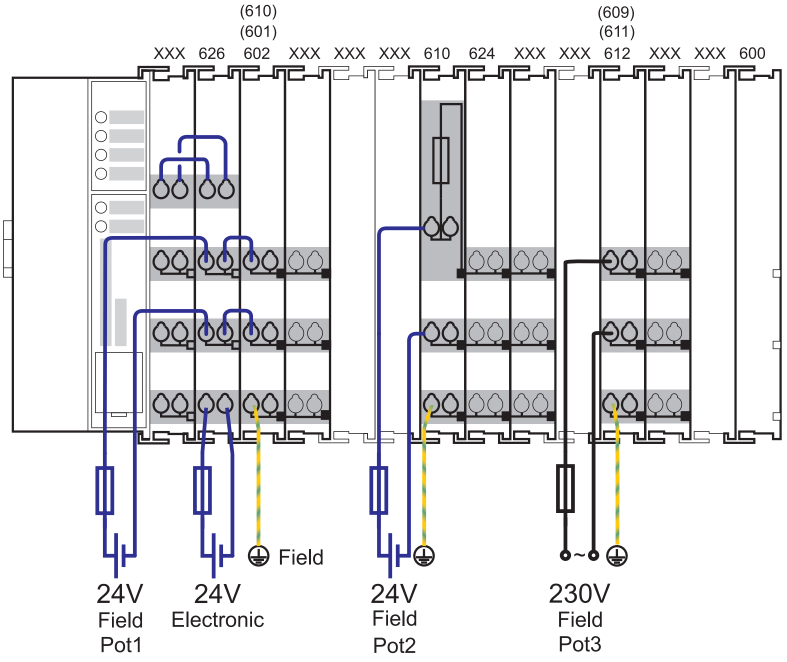


Fig. 2-21: Power supply concept g01xx11e

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

2.7.5 Supply example

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

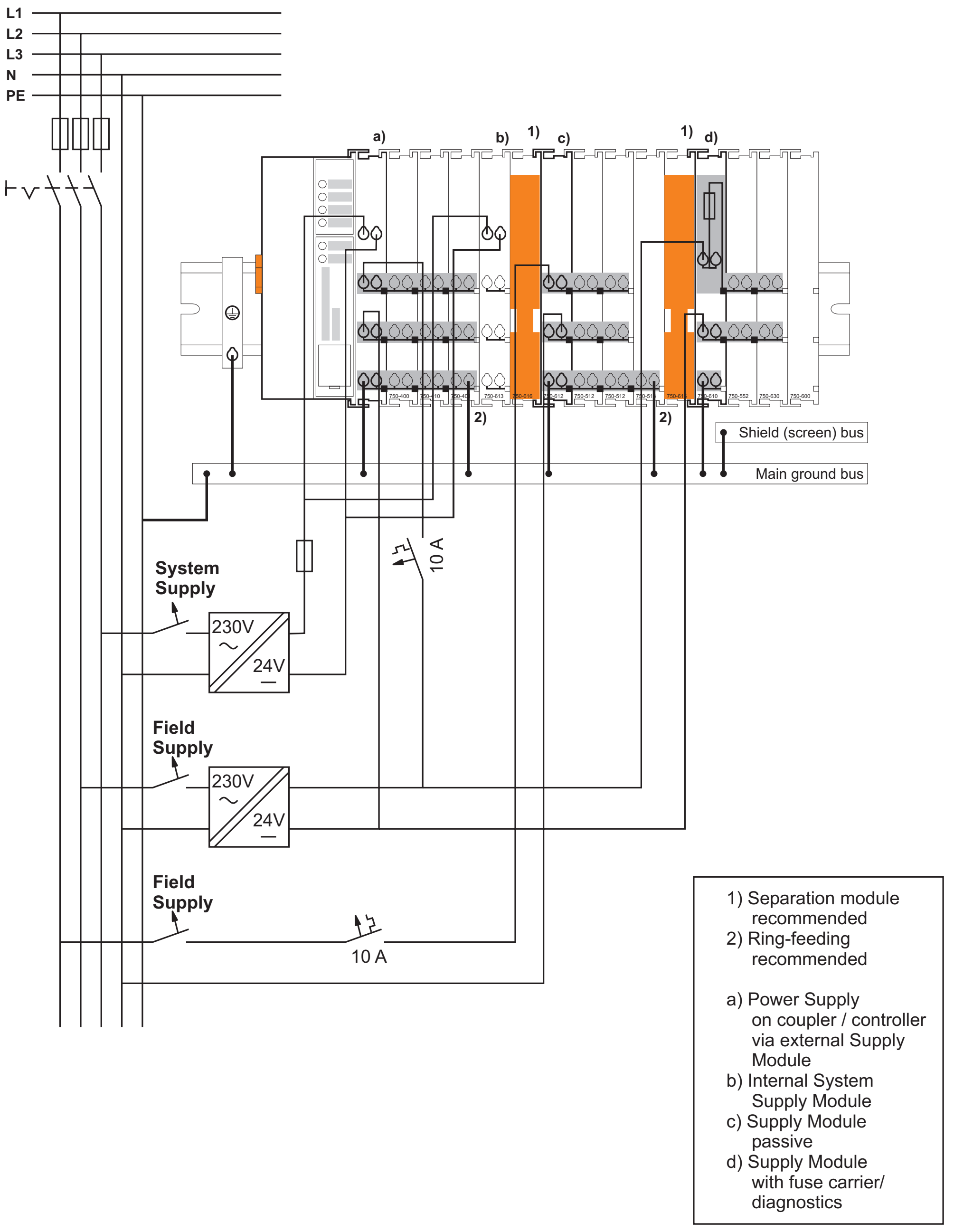


Fig. 2-22: Supply example

g0xxx04e

2.7.6 Power Supply Unit

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

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

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

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

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

WAGO products Article No.	Description
787-903	Primary switched - mode, DC 24 V, 5 A wide input voltage range AC 85-264 V PFC (Power Factor Correction)
787-904	Primary switched - mode, DC 24 V, 10 A wide input voltage range AC 85-264 V PFC (Power Factor Correction)
787-912	Primary switched - mode, DC 24 V, 2 A wide input voltage range AC 85-264 V PFC (Power Factor Correction)
	Rail-mounted modules with universal mounting carrier
288-809	AC 115 V / DC 24 V; 0,5 A
288-810	AC 230 V / DC 24 V; 0,5 A
288-812	AC 230 V / DC 24 V; 2 A
288-813	AC 115 V / DC 24 V; 2 A

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.

Article No.	Description
283-609	Single-conductor ground (earth) terminal block make an automatic contact to the carrier rail; conductor cross section: 0.2 -16 mm2 Note: Also order the end and intermediate plate (283-320)

2.8.2 Grounding Function

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

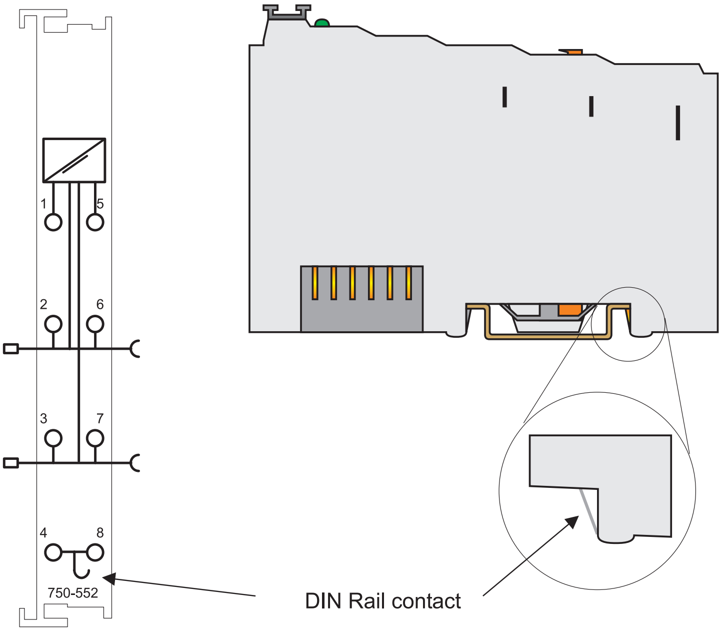


Fig. 2-23: Carrier rail contact g0xxx10e

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

The carrier rail must be grounded.

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

2.8.3 Grounding Protection

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

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

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

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

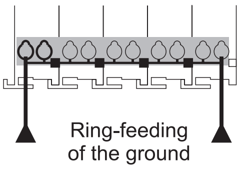


Fig. 2-24: Ring-feeding g0xxx07e

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

2.9 Shielding (Screening)

2.9.1 General

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

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

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

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

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

2.9.2 Bus Conductors

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

2.9.3 Signal Conductors

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

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

2.9.4 WAGO Shield (Screen) Connecting System

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

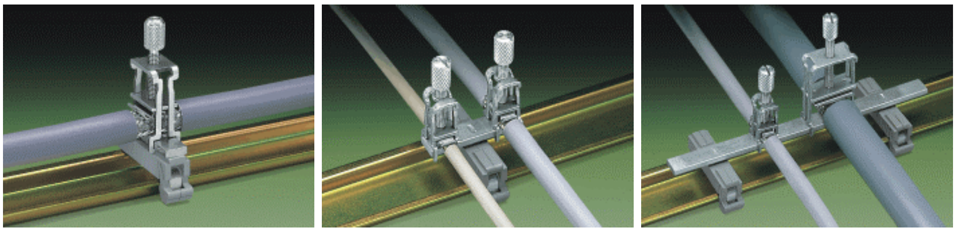


Fig. 2-25: WAGO Shield (Screen) Connecting System p0xxx08x, p0xxx09x, and p0xxx10x

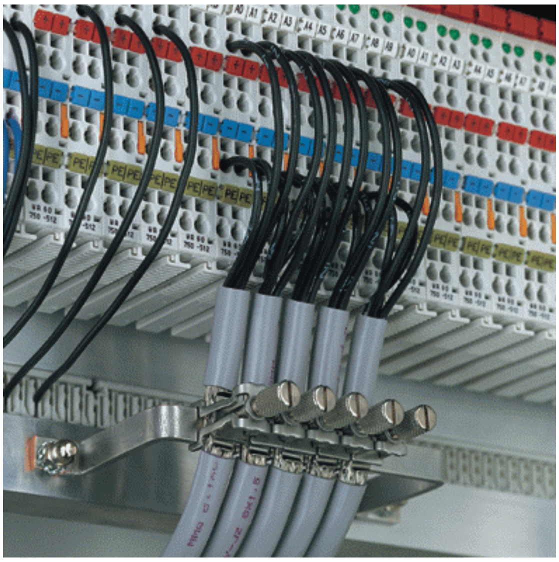


Fig. 2-26: Application of the WAGO Shield (Screen) Connecting System p0xxx11x

2.10 Assembly Guidelines/Standards

DIN 60204,	Electrical equipping of machines
DIN EN 50178	Equipping of high-voltage systems with electronic components (replacement for VDE 0160)
EN 60439	Low voltage – switch box combinations

3 Fieldbus Controller

3.1 Fieldbus Controller 750-841

3.1.1 Description

The WAGO 750-841 Programmable Fieldbus Controller (PFC) combines the functionality of an ETHERNET fieldbus coupler with the functionality of a Programmable Logic Controller (PLC). When the PFC is used as a PLC, all or some of its I/O modules can be control locally with the use of WAGO-I/O-*PRO* CAA. WAGO-I/O-PRO CAA is an IEC 61131-3 programming tool, which based on the standard programming system CoDeSys (a product of the company 3S) with specific addition of the target files for all WAGO controllers, used to program and configure the 750-841 PFC. I/O modules which are not controlled locally, can be controlled remotely through the 10/100 Mbps ETHERNET Fieldbus port.

When power is applied to the PFC, it automatically detects all I/O modules connected to the controller and creates a local process image. This can be a mixture of analog and digital modules. The process image is subdivided into an input and an output data area.

The data of the analog modules is mapped first into the process image. The modules are mapped in the order of their position after the controller. The digital modules are grouped after the analog modules, in the form of words (16 bits per word). When the number of digital I/O's exceeds 16 bits, the controller automatically starts another word.

The controller has 512 KB of program memory, 128 KB of data memory, and 24 KB of retained memory. The programmer has access to all fieldbus and I/O data.

To be able to send/receive process data via ETHERNET, the controller supports a series of network protocols. For the exchange of process data, the MODBUS TCP (UDP) protocol and the Ethernet/IP protocol are available. Both communication protocols can be used alternatively or parallel.
For this, the write authorization, that means the access by PLC, MODBUS/TCP or EtherNet/IP on the I/O modules are specified in a xml-file.

The protocol HTTP, BootP, DHCP, DNS, SNTP, FTP, SNMP and SMTP are provided for the management and diagnosis of the system.

The programmer has the option to use function modules for programming clients and servers for all transport protocols (TCP, UDP, etc.) via a socket-API.

Library functions are available to extend the range of programming functions. The IEC 61131-3 library "SysLibRTC.lib" enables integration of a buffered real time clock with date (1 second resolution), alarm function, and a timer. In the event of a power failure, this clock is powered by an auxiliary supply.

The controller is based on a 32-bit CPU and is capable of multitasking (i.e., several programs can be run at the same time).

The controller has an internal server for web-based applications. By default, the controller’s built-in HTML pages contain information on the configuration and status of the PFC, and can be read using a normal web browser. In addition, a file system is implemented that allows you to store custom HTML pages in the controller using FTP download.

3.1.2 Compatibility

Programming tool:	WAGO-I/O-*PRO* 32 759-332		WAGO-I/O-*PRO* CAA 759-333				
	V2.1	V2.2.6	V2.3.2.5	V2.3.2.7	V2.3.3.4	V2.3.3.6	V2.3.4.3
Controller:							
750-841	-	-	✓	✓	✓	SW ≥ 06	SW ≥ 09

-	Controller NOT compatible with WAGO-I/O-PRO version
✓	Controller compatible with WAGO-I/O-PRO version, independent of the controller hard- or software
SW ≥ xy	Controller compatible with WAGO-I/O-PRO version if the controller has software xy or higher

Attention
The CoDeSys network variables from WAGO-I/O-PRO V2.3.3.6 and higher are supported by the controllers 750-841 with the software SW ≥ 06.
The WEB visualisation from WAGO-I/O-PRO V2.3.4.3 and higher are supported by the controllers 750-841 with the software SW ≥ 09.

3.1.3 Hardware

3.1.3.1 View

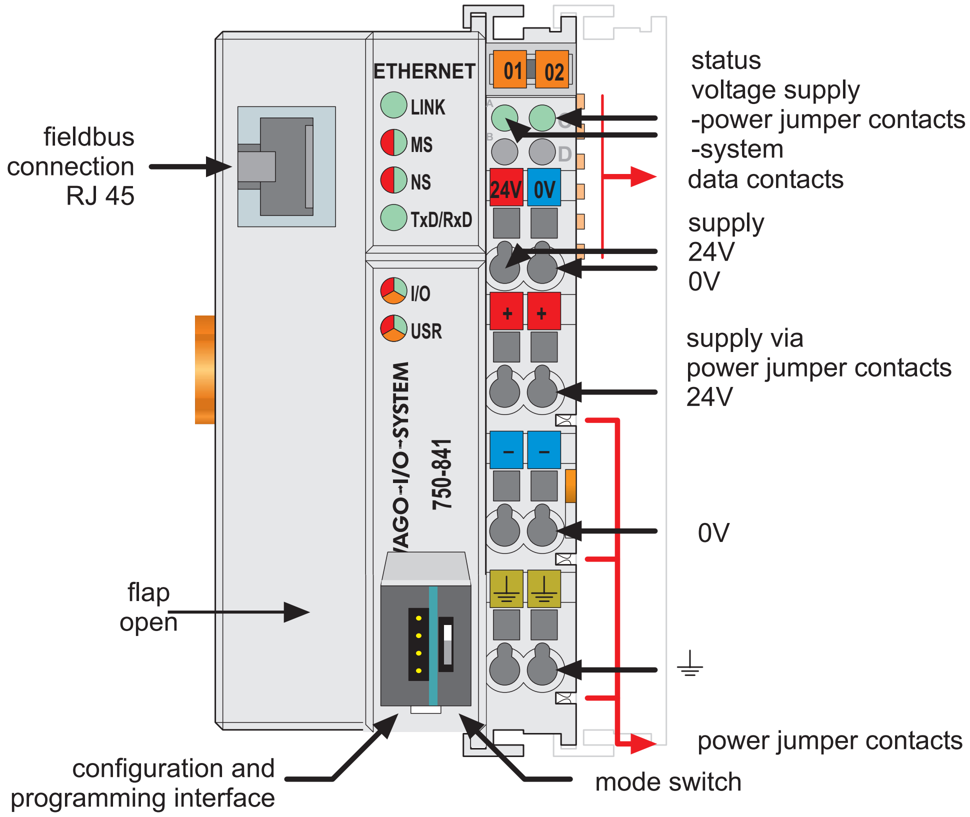


Fig. 3.1-1: Fieldbus controller ETHERNET TCP/IP g084100e

The Fieldbus Controller consists of:

- Device supply with internal system supply module for the system supply as well as power jumper contacts for the field supply via assembled I/O modules
- Fieldbus interface with the bus connection
- Display Elements (LED's) for operation status, diagnostics, and communication status
- Configuration and programming interface port
- Operating mode switch
- Electronics for communication with the I/O modules (internal bus) and the fieldbus interface

3.1.3.2 Device Supply

The PFC is powered via terminal blocks with CAGE CLAMP® connections. The Device Supply generates the necessary voltages to power the electronics of the controller and the internal electronics of the connected I/O modules.

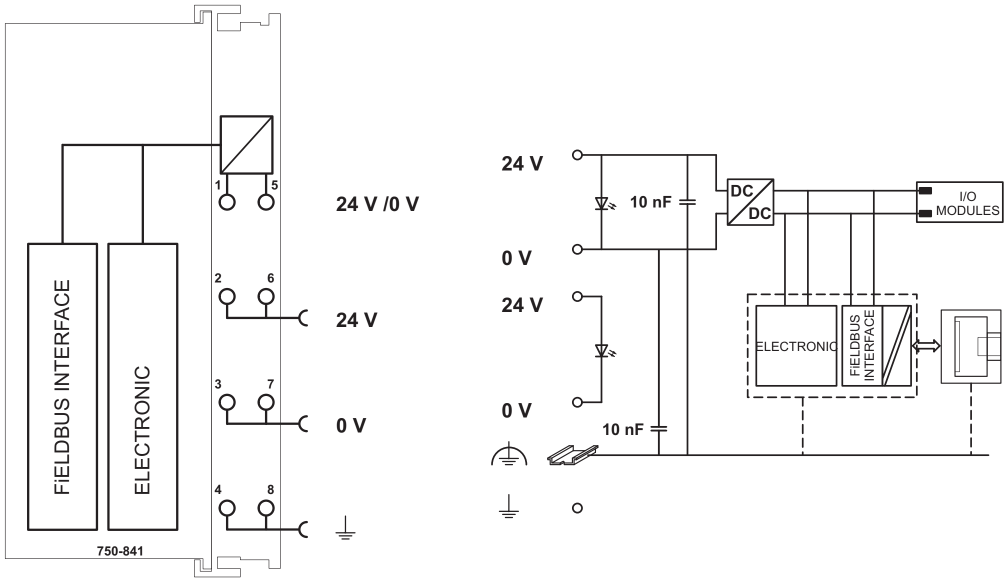


Fig. 3.1-2: Device supply G084101e

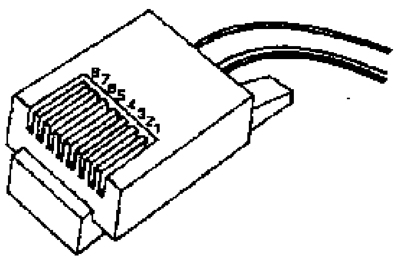
The internal electronics of the controller and I/O modules are electrically isolated from the field-side power connections and field devices by the use of DC/DC converters and optocouplers.

3.1.3.3 Fieldbus Connection

Connection to the fieldbus is by a RJ45 connector. The RJ45 socket on the fieldbus controller is wired per the 100BaseTX standard. The specification for the connecting cable is a twisted pair cable of Category 5. Cables of type S-UTP (Screened-Unshielded Twisted Pair) and STP (Shielded Twisted Pair) with a maximum segment length of 100 meters may be used.

The RJ45 socket is physically lowered for the controller to fit in an 80 mm high switch box once connected.

The electrical isolation between the fieldbus system and the electronics is achieved by DC/DC converters and optocouplers in the fieldbus interface.



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

Fig. 3.1-3: RJ45-Connector and RJ45 Connector Configuration

Attention!
Only for use in LAN, not for connection to telecommunication circuits!

3.1.3.4 Display Elements

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

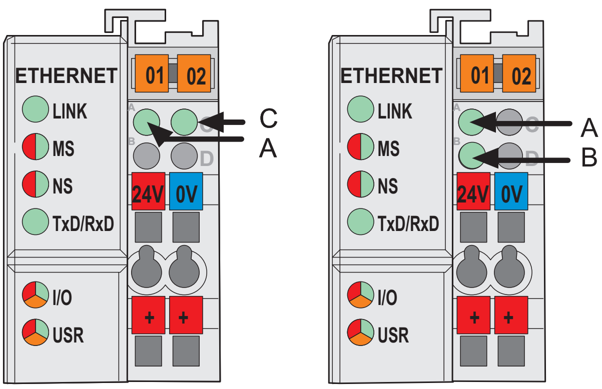


Fig. 3.1-1: Display Elements 750-841
g084102x

LED	Color	Meaning
LINK	green	Link to a physical network exists
MS	red/green	The ‚MS‘-LED indicates the state of the node (Module State)
NS	red/green	The ‚NS‘-LED indicates the state of the network (Network State)
TxD/RxD	green	Data exchange taking place
IO	red /green / orange	The 'I/O'-LED indicates the operation of the node and signals faults encountered
USR	red /green / orange	The 'USR' LED can be controlled by a user program in a controller
A	green	Status of the operating voltage – system
B or C	green	Status of the operating voltage – power jumper contacts (LED position is manufacturing dependent)

More Information
The evaluation of the displayed LED signals is described in Chapter 3.1.11 "LED Display".

3.1.3.5 Configuration and Programming Interface

The Configuration and Programming Interface port is located behind a cover flap. This communications port can be use with WAGO-I/O-CHECK and WAGO-I/O-PRO CAA, as well as for firmware downloading.

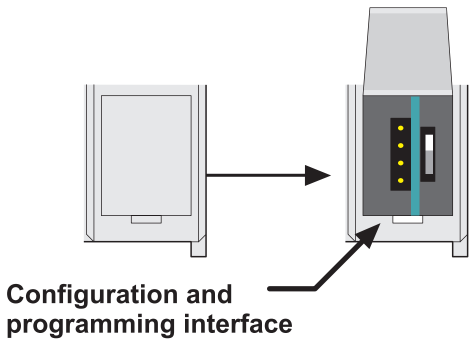


Fig. 3.1-2: Configuration and Programming Interface g01xx07e

A WAGO 750-920 Communication Cable is used to connect the 4 pin male header and with a PC’s 9-pin RS232 interface.

Warning
The communication cable 750-920 must not be connected or disconnected while the coupler/controller is powered on!

3.1.3.6 Operating Mode Switch

The operating mode switch is located behind a cover flap.

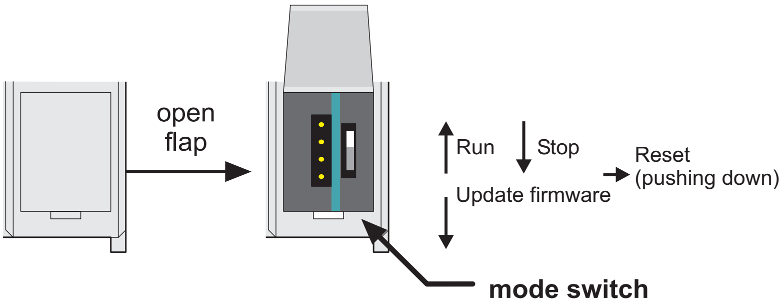


Fig. 3.1-4: Operating Mode Switch g01xx10e

The switch is a push/slide switch with 3 settings and a hold-to-run function.

The slide switch is designed for a maximum number of switching cycles as defined in EN61131T2.

Operating mode switch	Function
From middle to top position	Firmware and PFC application are executed (Activate program processing (RUN)
From top to middle position	Firmware is executed, PFC application is stopped (Stop program processing (STOP)
Lower position, bootstrap	Controller starts the operating system loader
Push down (i.e. with a screwdriver)	Hardware reset All outputs and flags are reset; variables are set to 0 or to FALSE or to an initial value. Retain variables or flags are not changed. The hardware reset can be performed with STOP as well as RUN in any position of the operating mode switch!

An operating mode (i.e., RUN/STOP) is internally changed at the end of a PLC cycle.

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

Attention
If outputs are set when switching from RUN to STOP mode, they remain set! Switching off the outputs on the software side i.e. by the initiators are ineffective because the program is no longer processed.

Note
The user has the possibility to define the status of the outputs before a STOP condition. For this, in the web-based management system a web-page opens via the PLC link, on that the function can be accordingly specified.
If there is a checkmark in the small box behind "Enabled", all outputs are set to zero, otherwise the outputs remain to the last current value.

3.1.3.7 Hardware Address (MAC-ID)

Each WAGO ETHERNET TCP/IP fieldbus controller is supplied from the factory with a unique and internationally unambiguous physical ETHERNET address, also referred to as 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 controller's side. The address has a fixed length of 6 Bytes (48 Bits) and contains the address type, the manufacturer's ID, and the serial number.

3.1.4 Operating System

3.1.4.1 Start-up

The controller starts-up after switching on the supply voltage or after a hardware reset.

Note
The Operating Mode slide switch must not be in the bottom position during start-up!

The PLC program in the flash memory is transferred to RAM.

This is followed by the initialization of the system. The controller determines the I/O modules and the present configuration. The variables are set to 0, FALSE, or to an initial value given by the PLC program. The flags retain their status . The "I/O" LED blinks red during this phase.

Following a fault free start-up the controller changes over to "RUN" mode. The "I/O" LED lights up green.

3.1.4.2 PLC Cycle

The PLC cycle starts following a fault free start-up when the Operating Mode Switch is in the top position or by a start command from the WAGO-I/O-*PRO* CAA. The controller starts a PLC cycle by first reading the fieldbus data, I/O modules, and time data. Next, the PLC program in RAM is processed (scanned). After the program is processed, the fieldbus data and I/O modules are updated with new output data. System functions are then preformed (i.e., system diagnostics, communications, time calculations, etc). At this point, if a STOP command is not present, the cycle starts over again with the reading of the fieldbus data, I/O modules, and time data.

The change of the operating mode (STOP/RUN) is only made at the end of a PLC cycle.

The cycle time is the time from the start of the PLC program to the next start. If a sizeable loop is programmed within a PLC program, the PLC cycle time is extended correspondingly.

The inputs and outputs are not updated during the scanning of the PLC program. I/O updates only occur at the end of the PLC program scan. For this reason, it is not possible to wait for a physical I/O change from within a program loop.

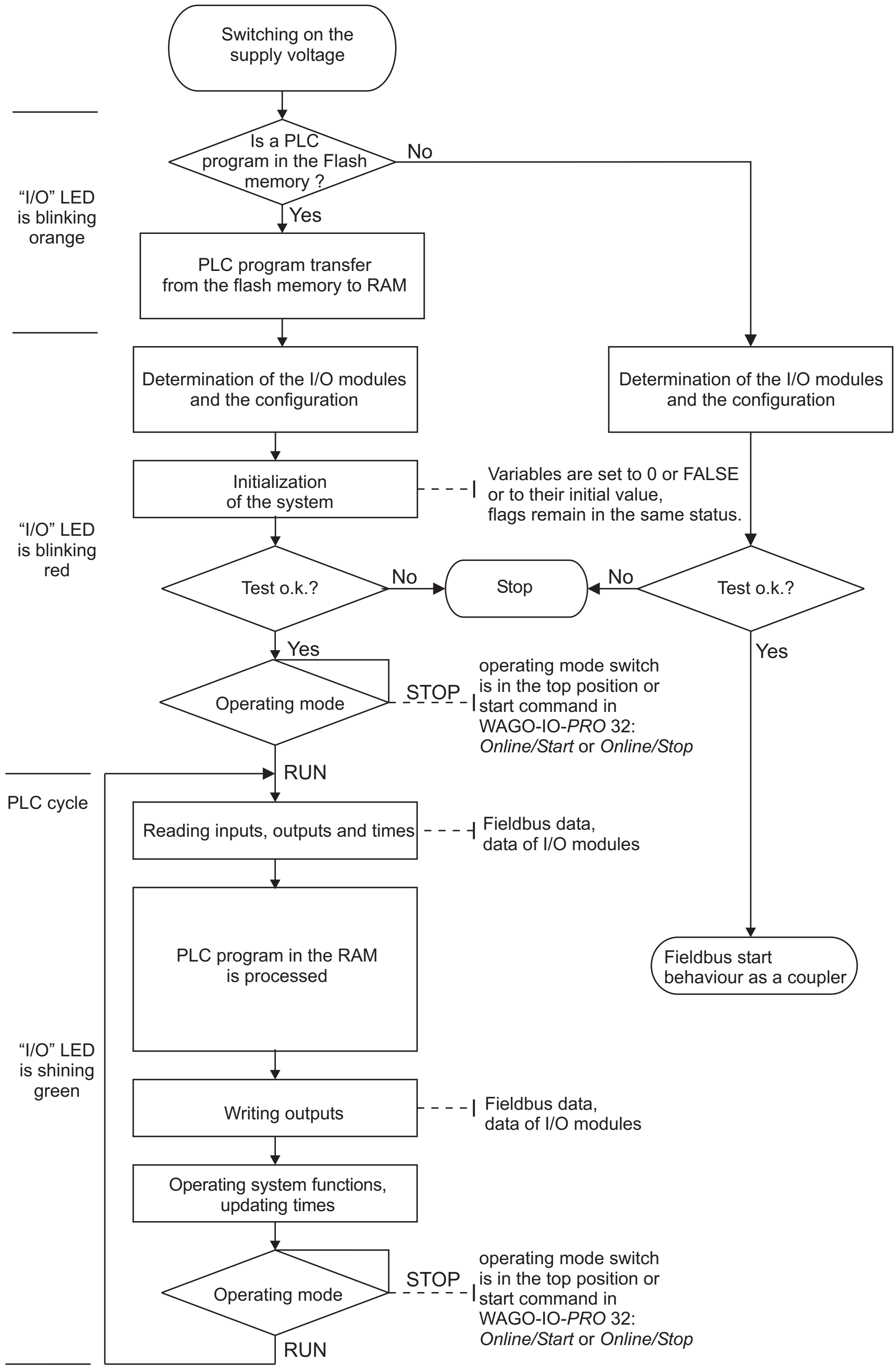


Fig. 3.1-5: Controller Operating System g012941e

3.1.5 Process Image

3.1.5.1 General Structure

The powered-up controller recognizes all I/O modules connected in the node that are waiting to transmit or receive data (data width/bit width > 0). The maximal length of a node is limited to 64 I/O modules.

Note
Use of the WAGO 750-628 Bus Extension Coupler Module and the 750-627 Extension End Module enables support of up to 250 I/O modules on the 750-841 controller.

Note!
Expansion to 250 I/O modules is enabled in the controllers with software version ≥ SW 9.

Note
For the number of input and output bits or words of an individual I/O module, please refer to the corresponding module description later in this chapter.

The controller generates an internal local process image from the data width and type of I/O modules, as well as the position of the I/O modules in the node. This image is divided into an input and an output area.

The data of the digital I/O modules are bit-based (i.e., the data exchange is made by bits). The analog I/O and most specialty modules (e.g., counter modules, encoder modules, and communication modules) are byte-based, in which the data exchange is made by bytes.

The process image is divided into an input and an output data area. Each I/O module is assigned a location in the process image, based on the data exchange type (i.e., bit-based or byte-based) and their position after the controller.

All of the byte-based I/O modules are filled in the process image first, then the bit-based modules. The bits of the digital modules are grouped into a word. Once the number of digital I/Os exceeds 16 bits, the controller automatically starts another word.

Note
Changing the physical layout of a node will result in a new structure of the process image. Also, the addresses of the process data will change. When adding or removing modules, the process data must be verified.

The process image for physical input and output data is stored in the first 256 words of memory (word 0 to 255). This memory actually consists of a separate area for the input and output data, but both areas are referenced in a PLC program with an index of 0 to 255 for word operations.

The MODBUS PFC variables are mapped after the process image of the I/O modules. This memory area contains 256 words (word 256 to 511).

If the quantity of I/O data is greater than 256 words, the additional data is appended after the MODBUS PFC variables in word 512 to 1275. Like the first physical I/O process image area, there is a separate memory area for input and output data, but both are referenced with an index of 512 to 1275 for word operations.

After the remaining physical I/O data is the Ethernet IP PFC variables. This memory area is word 1276 to 1531.

For future protocol additions, the area above word 1532 is reserved for additional PFC variables.

With all WAGO fieldbus controllers, the method used by PLC functions to access process data is independent of the fieldbus system. This access always takes place via an application-related IEC 61131-3 program.

In contrast to the above, access from the fieldbus side is fieldbus specific. For the ETHERNET TCP/IP fieldbus controller, either a MODBUS/TCP master or an Ethernet/IP master is used. MODBUS/TCP accesses the data via implemented MODBUS functions. Here decimal and/or hexadecimal MODBUS addresses are used. With Ethernet/IP, data access occurs with the use of an object model.

More Information
A detailed description of these fieldbus-specific data access operations is given in the sections “MODBUS functions” and “Ethernet/IP (Ethernet/Industrial Protocol)”.

More Information
You can find the fieldbus specific process data architecture for all I/O Modules in the chapter „Fieldbus specific Process Data Architecture“.

3.1.5.2 Example of a Process Input Image

The following figure is an example of a process input image. The configuration includes 16 digital and 8 analog inputs. Therefore, the process image has a total data length of 9 words (8 words for the analog data and 1 word for the digital inputs).

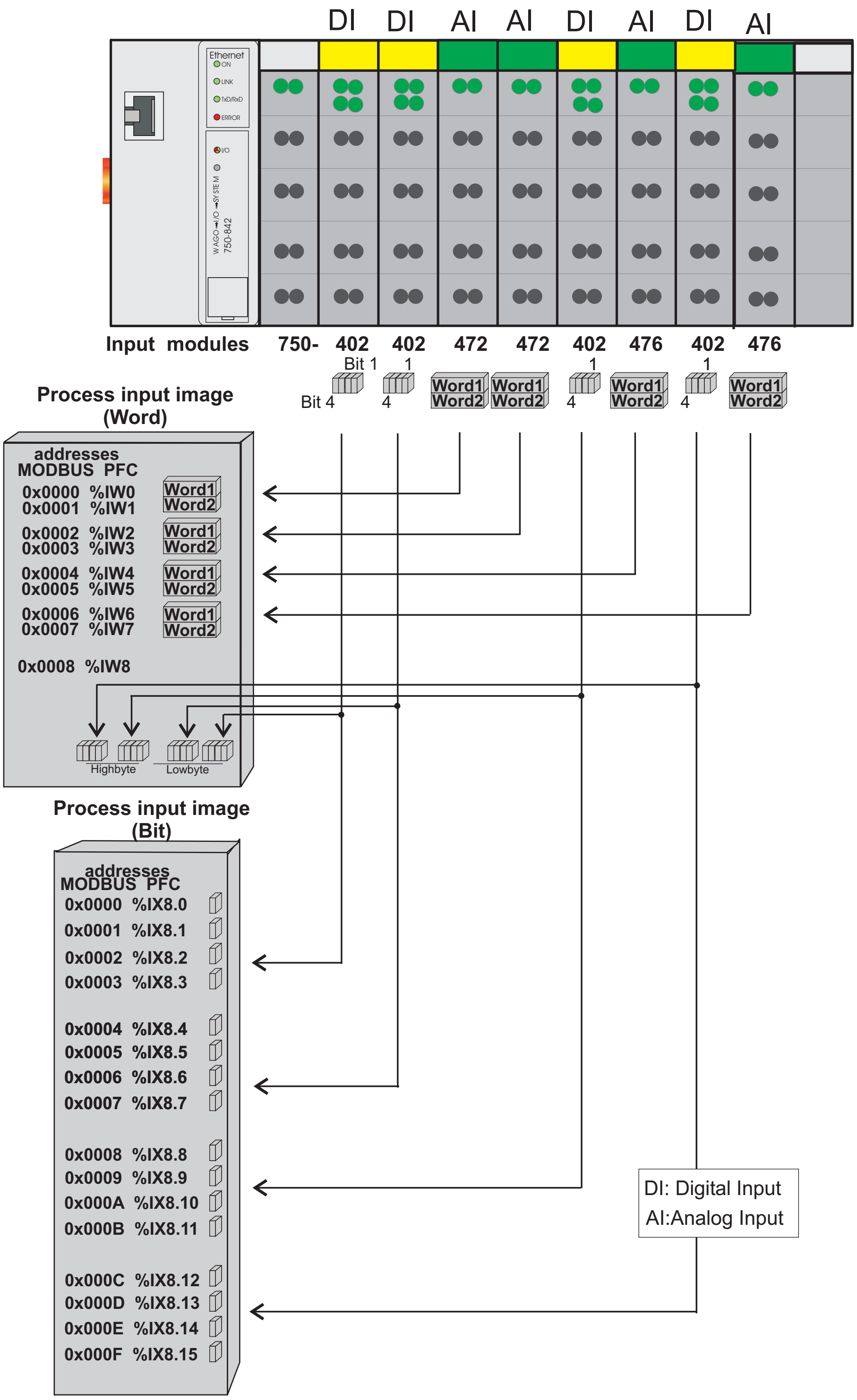


Fig. 3.1-6: Example of a Process Input Image G012924e

3.1.5.3 Example of a Process Output Image

The following figure is an example of a process output image. The configuration includes 2 digital and 4 analog outputs. Therefore, the process image has a total data length of 5 (4 words for the analog data and 1 word for the digital outputs). When using MODBUS protocol, output data can be read back with an offset of 200_{hex} (0x0200) added to the MODBUS address.

Note
All output data over 256 words can be read back with an offset of 1000_{hex} (0x1000) added onto the MODBUS address.

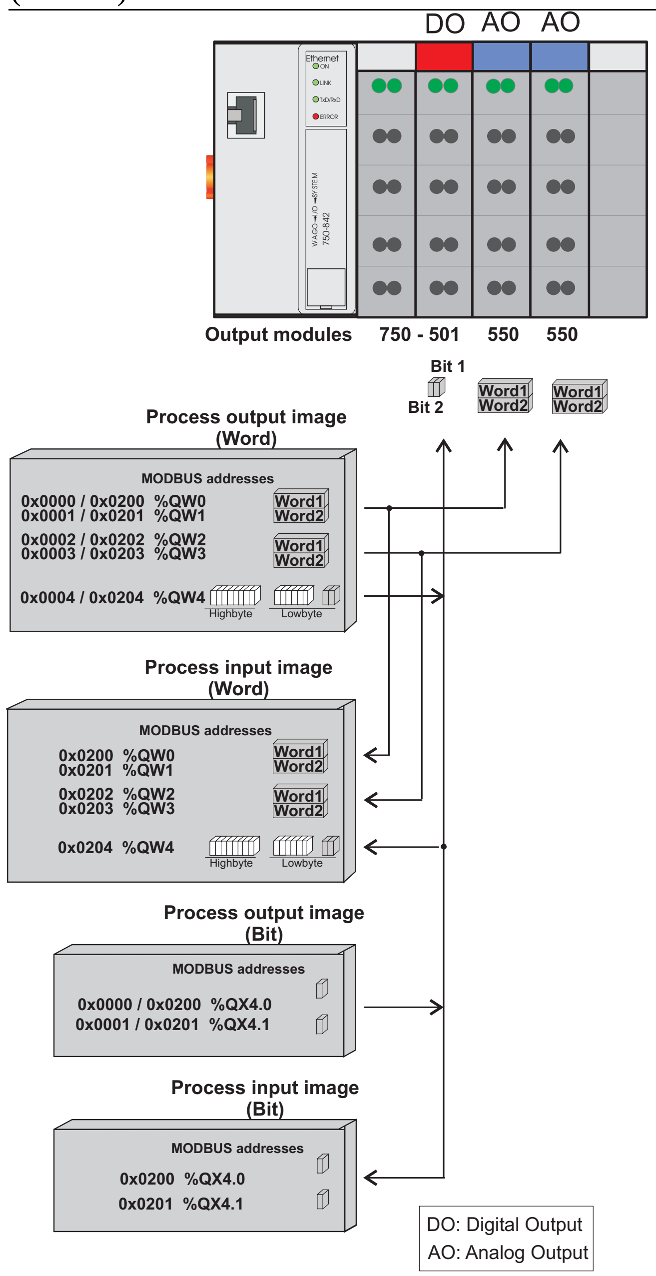


Fig. 3.1-7: Example of a Process Output Image G012925e

3.1.5.4 Process Data Architecture

With some I/O modules, the structure of the process data is fieldbus specific. In the case of an Ethernet TCP/IP coupler/controller, 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.

More Information
You can find the fieldbus specific process data architecture for all I/O Modules of the WAGO-I/O-SYSTEM 750 and 753 in the chapter „Process Data Architecture“.

3.1.6 Data Exchange

The ETHERNET TCP/IP fieldbus controller can be configured for either MODBUS/TCP or the Ethernet IP protocol.

MODBUS/TCP works according to the master/ model. The master (e.g., a PC or a PLC) will query a slave device and the slave will return a response to the master depending on the kind of query. Queries are addressed to a specific node through the use of the IP address.

Typically, the ETHERNET TCP/IP controller of the WAGO-I/O-SYSTEM is a slave device. But, with the use of the WAGO-I/O-*PRO* CAA programming tool, the PFC can additionally perform master functions.

A controller is able to produce a defined number of simultaneous socket connections to other network subscribers:

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

The maximum number of simultaneous connections may not be exceeded. If you wish to establish further connections, terminate an existing connection first.
For data exchange, the ETHERNET TCP/IP fieldbus controller uses three main interfaces:

- interface to the fieldbus (master),
- the PLC functionality of the PFCs (CPU) and
- the interface to the I/O Modules.

Data exchange takes place between the fieldbus master and the I/O modules, between the PLC functionality of the controller and the I/O modules as well as between the fieldbus master and the PLC functionality of the controller. Currently, the 750-841 supports MODBUS/TCP and ETHERNET IP based master devices. When the controller performs PLC functions, and controls various I/O modules, this is done with the use of an IEC 61131-3 application program, whereby the data addressing is different than the fieldbus addressing.

3.1.6.1 Memory Areas

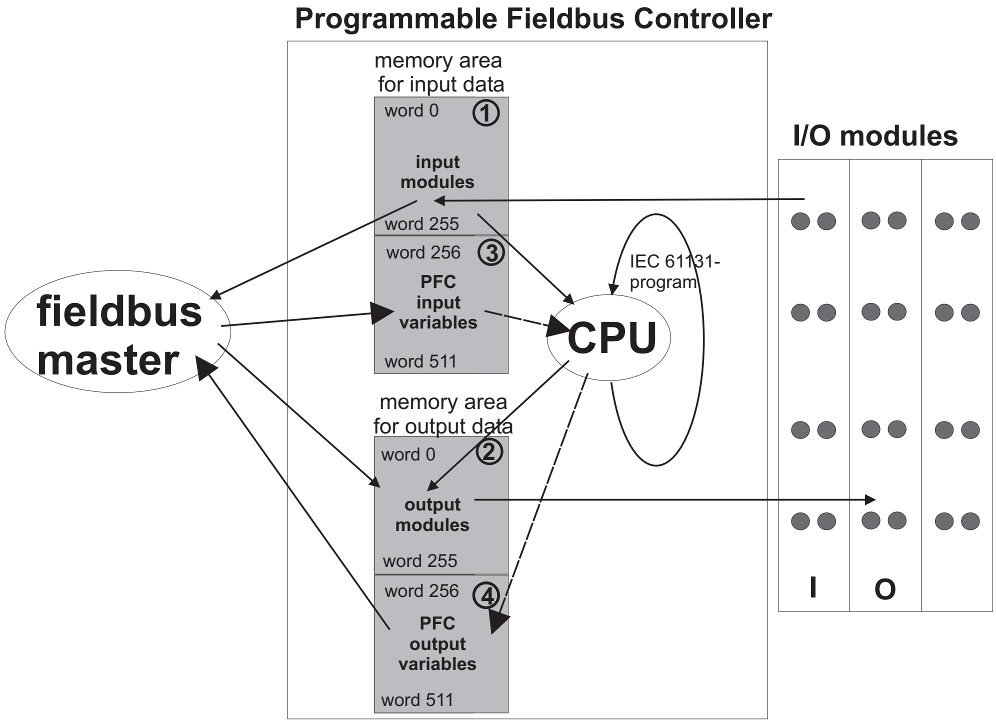


Fig. 3.1-8: Memory Areas and Data Exchange for a Fieldbus Controller g012938e

The PFC‘s process image contains the physical data of the I/O modules in memory words 0 to 255 and 512 to 1275.

(1) Reading data of the input modules is possible from both the controller’s CPU and from the fieldbus master (See Figure 3-8).

(2) In the same manner, writing data to output modules is possible from both the controller’s CPU and from the fieldbus master.

The controller’s process image also contains variables called “PFC Variables”. These variables are allocated based on the fieldbus protocols. The MODBUS TCP PFC variables are stored in memory from word 256 to 511. Ethernet IP PFC variables are stored in memory from word 1276 to 1531. The memory area above word 1531 is reserved for future protocols.

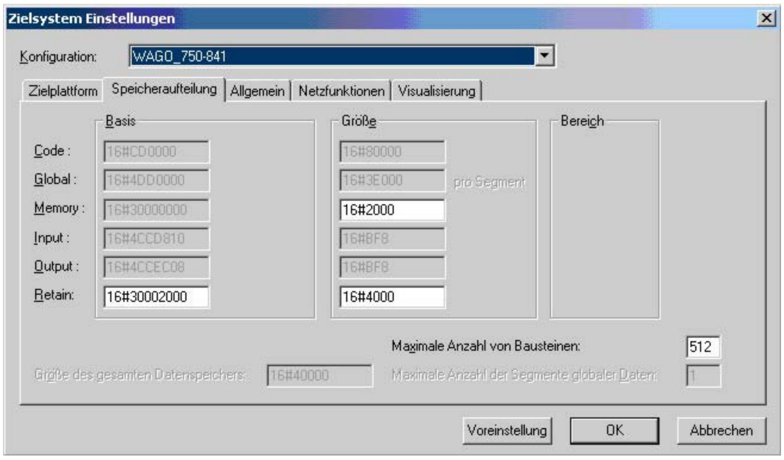
(3) The PFC input variables are written into the input memory space from the fieldbus master and can be read by the controller’s CPU for further processing.

(4) The variables processed by the controller’s CPU , via an IEC 61131-3 application program, can be written to the PFC Variables and then read by the fieldbus master.

In addition, with the MODBUS TCP/IP protocol, all output data has a mirrored image in memory with the address offset 0x0200 or 0x1000. This permits reading back output values after they are written by adding 0x0200 or 0x1000 to the MODBUS address.

In addition, the controller offers other memory spaces which partly cannot be accessed from the fieldbus master:

<table>
<tr><td>RAM
Memory
256 kByte</td><td>The RAM memory is used to create variables not required for communication with the interfaces but for internal processing, such as for instance computation of results.</td></tr>
<tr><td>Code-
Memory
512 kByte</td><td>The IEC 61131-3 program is filed in the code memory. The code memory is a flash ROM. Once the supply voltage is applied, the program is transmitted from the flash to the RAM memory.
After a successful start-up, the PFC cycle starts when the operating mode switch is turned to its upper position or by a start command from WAGO-I/O-PRO CAA.</td></tr>
<tr><td>NOVRAM
Remanent
Memory
24 kByte

(Retain)</td><td>The remanent memory is non volatile memory, i.e. all values are retained following a voltage failure. The memory management is automatic. This 24 kByte sized memory area (word 0 ... 12288) devides on into a 8 kByte sized addressable range for the flags (%MW0... %MW4095) and a 16 kByte sized Retain range for variables without memory space addressing or variables which are explicitly defined with "var retain".
Note
The allocation of the NORAM is if necessary changeable in the programming software WAGO-I/O-PRO CAA/
register: "Ressourcen"/dialog: "Target Settings" (see picture).


The start address for the flag range is thereby firmly addressed with 16#30000000. The range sizes and the start address of the Retain memory are variable. To exclude a data overlap of the ranges, it is however recommended, to maintain the default settings. Thereby the size of the flag range is given with 16#2000 and subsequently to it the Retain memory has the start address 16#30002000 and the size of 16#4000.</td></tr>
</table>

3.1.6.2 Addressing

3.1.6.2.1 Addressing the I/O Modules

The arrangement of the I/O modules in a node is flexible and up to the user. Although, the user must verify that the power jumper contacts from one I/O module to the next are compatible and at the same voltage level.

When the controller addresses I/O modules, data of complex modules (modules occupying 1 or more bytes) are mapped first. They are mapped in the order of their physical position after the controller. As such, they occupy the addresses beginning with word 0. Following this, the digital modules are grouped in the form of words (16 bits per word). They are also arranged by their physical order. When the number of digital I/O's exceeds 8 bits, the controller automatically starts another byte.

Note
For detailed information on the number of input and output bits/bytes of a specific module, please refer to the modules manual.

Note
Changing the physical layout of a node will result in a new structure of the process image. Also, the addresses of the process data will change. When adding or removing modules, the process data must be verified.

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 thermal elements	Digital output modules with diagnosis (2 Bit / channel)
Input modules for resistance sensors	Power supply modules with fuse holder / diagnosis
Pulse width output modules	Solid State power relay
Interface module	Relay output modules
Up/down counter	
I/O modules for angle and path measurement	

Table 3.1.1: I/O Module Data Width

3.1.6.2.2 Address Range

Partition of Address ranges for the word-wise addressing acc. to IEC 61131-3 :

Word	Data
0-255	physical I/O modules
256-511	MODBUS/TCP PFC variables
512-1275	remaining physical I/O modules
1276-1531	Ethernet/IP PFC variables
1532-.....	reserved for PFC variables of future protocols

Word 0-255: First address range I/O module data:

Data	Address								
Bit	0.0 ... 0.7	0.8... 0.15	1.0 ... 1.7	1.8... 1.15	.....	254.0 ... 254.7	254.8... 254.15	255.0 ... 255.7	255.8... 255.15
Byte	0	1	2	3	.....	508	509	510	511
Word	**0**		**1**		**.....**	**254**		**255**	
Dword	0				.....	127			

Table. 3.12: First Address Range for the I/O Module Data

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

Data	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			

Table 3.1.3: Address Range for the MODBUS/TCP Fieldbus Data

Word 512-1275: Second address range I/O module data:

Data	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			

Table 3.1.4: Second Address Range for the I/O Module Data

Word 1276-1531: Address range for Ethernet/IP fieldbus data:

Data	Address								
Bit	1276.0 ... 1276.7	1276.8 ... 1276.15	1277.0 ... 1277.7	1277.8 ... 1277.15	...	1530.0 ... 1530.7	1530.8 ... 1530.15	1531.0 ... 1531.7	1531.8 ... 1531.15
Byte	2552	2553	2554	2555	...	3060	3061	3062	3063
Word	**1276**		**1277**		**...**	**1530**		**1531**	
Dword	638				...	765			

Table 3.1.5: Address Range for the Ethernet IP Fieldbus Data

Address range for flags (Retain Variables):

Data	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			

Table 3.1.6: Address Range for Flags (Retain Variables)

Overview of the IEC 61131-3 address ranges:

Address range	MODBUS Access	SPS 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 SPS Input variables (%IW256 ... %IW511)
MODBUS/TCP PFC OUT variables	read	read/write	Volatile SPS Output variables (%QW256 ... %QW511)
Ethernet/IP PFC IN variables	-	read	Volatile SPS Input variables (%IW1276 ... %IW1531)
Ethernet/IP PFC OUT variables	-	read/write	Volatile SPS Output variables (%QW1276 ... %QW1531)
Configuration register	read/write	---	see Chapter „Ethernet“
Firmware register	read	---	see Chapter „Ethernet“
Flags/RETAIN variables	read/write	read/write	Remanent memory (%MW0 ... %MW12288)

Table 3.1.7: Overview IEC 61131-3 Address ranges

3.1.6.2.3 Absolute Addresses

Accessing individual memory cells (absolute addresses) in accordance with IEC 61131-3 is made using special character defined in the table below:

Position	Character	Designation	Comments
1	%	Starts 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	
e. g. word wise: %QW27 (28. Word), bit wise: %IX1.9 (10. Bit in Word 2)			
* The character ‘X’ for bits can be deleted			

Table 3.1.8: Absolute Addresses

Note
Enter the absolute address character strings without blanks (white spaces)!

Addressing Example:

	Inputs:			
Bit	%IX14.0 ... 15		%IX15.0 ... 15	
Byte	%IB28	%IB29	%IB30	%IB31
Word	**%IW14**		**%IW15**	
DWord	%ID7			

	Outputs:			
Bit	%QX5.0 ... 15		%QX6.0 ... 15	
Byte	%QB10	%QB11	%QB12	%QB13
Word	**%QW5**		**%QW6**	
DWord	%QD2 (upper section)		%QD3 (lower section)	

	Flags:			
Bit	%MX11.0 ... 15		%MX12.0 ... 15	
Byte	%MB22	%MB23	%MB24	%MB25
Word	**%MW11**		**%MW12**	
DWord	%MD5 (upper section)		%MD6 (lower section)	

Address calculation (depending upon the word address):

Bit address: word address .0 to .15
Byte address: 1. Byte: 2 x word address
2. Byte: 2 x word address + 1

Dword address:
word address (even numbers) / 2
or word address (uneven numbers) / 2, rounded off

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

The data exchange between the MODBUS/TCP Master and the I/O modules is made via the Ethernet Fieldbus port using MODBUS TCP read and write commands.

The controller handles four different types of process data with MODBUS TCP:

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

The relationship between bits and words are defined in the table below:

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							

Table 3.1.9: Allocation of Digital Inputs/Outputs to Process Data Word in Intel Format

Adding an offset of 0x0200 to the MODBUS output address lets you read back output data.

Note
For MODBUS mapping, all output data over 256 words resides in the memory area 0x6000 to 0x62FC, and can be read back with an offset of 1000_{hex} (0x1000) added onto the MODBUS address.

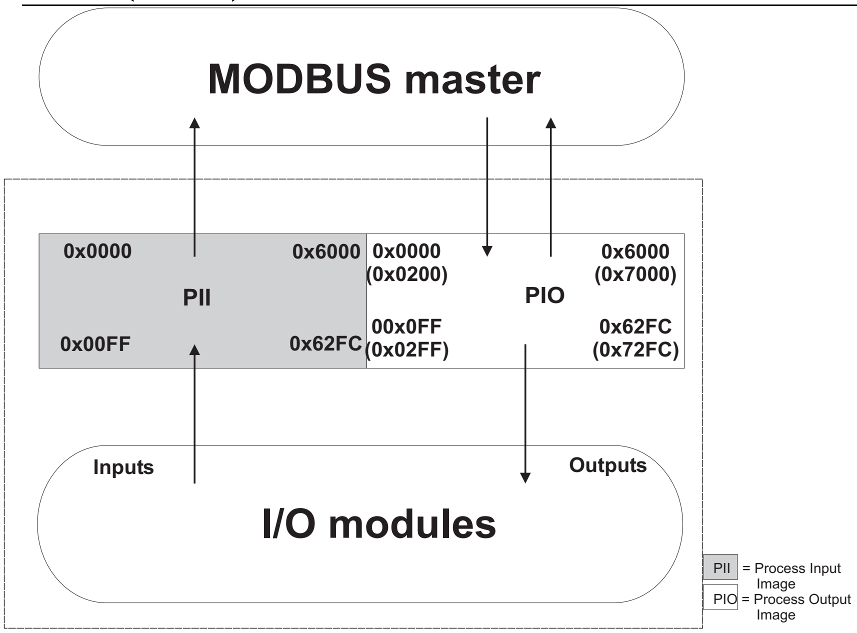


Fig. 3.1-9: Data exchange between MODBUS/TCP master and I/O modules g015045e

Starting from address 0x1000 there are the register functions. The register functions made available in the coupler, can be addressed by the MODBUS master along with the implemented MODBUS function codes (read/write). To this effect, the individual register address is entered in place of the address of a module channel.

More information
You can find a detailed description of the MODBUS addressing in the chapter „MODBUS Register Mapping“.

3.1.6.4 Data Exchange between EtherNet/IP Master and I/O Modules

Data exchange between the EtherNet/IP Master and the I/O modules is object oriented. Each node in the network is represented as a collection of objects. The "assembly" object defines the structure of objects for data transfer. With the assembly object, data (e.g. I/O data) can be grouped into blocks (mapped) and sent via a single communication link. As a result of this mapping technique, fewer access operations to the network are required.
Input and output assemblies have different functions.
An input assembly reads data from the application over the network or produces data on the network.
Where as, an output assembly writes data to the application or consumes data from the network.

Various assembly instances are permanently pre-programmed in the fieldbus controller (static assembly).

After switching on the power supply, the assembly object maps data from the process image. As soon as a connection is established, the master can address the data with "class", "instance" and "attribute" and access or read and/or write the data via I/O links.
The mapping of the data depends on the chosen assembly instance of the static assembly.

Further information
The assembly instances for the static assembly are described in the section "EtherNet/IP".

3.1.6.5 Data Exchange between PLC Functionality (CPU) and I/O Modules

Through absolute addresses, the PLC functionality of the controller can directly address the I/O module data.

The PFC addresses the input data with absolute addresses. The data can then be processed, internally in the controller, through the IEC 61131-3 program, whereby the flags are filed in a permanent memory area. Following this, the linking results can be directly written in the output data via absolute addressing.

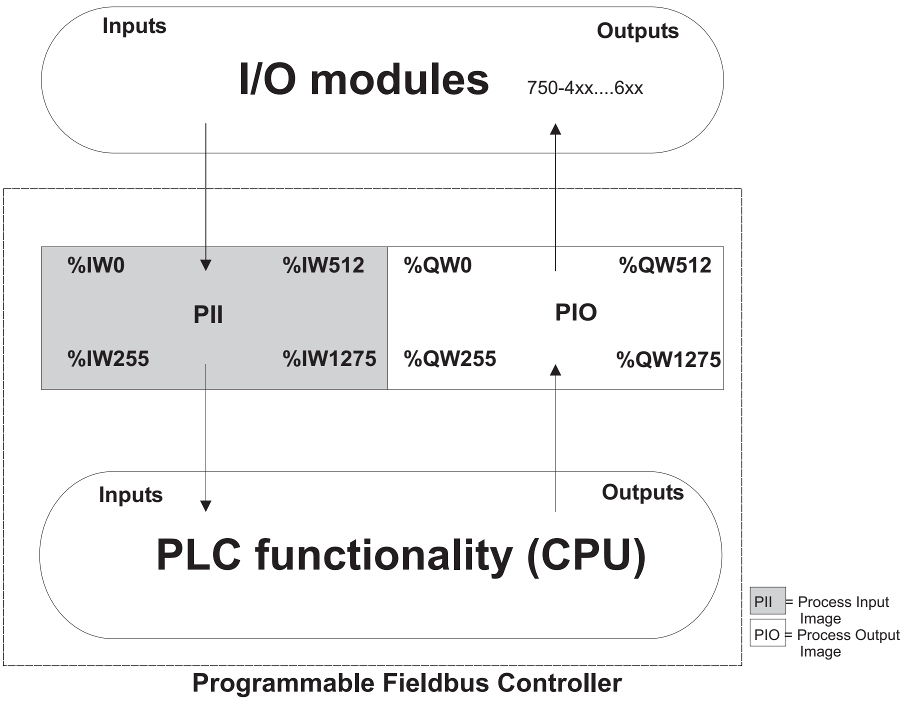


Fig. 3.1-10: Data exchange between PLC functionality (CPU) and I/O modules 15043e

3.1.6.6 Data Exchange between Master and PLC Functionality (CPU)

The fieldbus master and the PLC functionality of the controller regard the data in a different manner.

Variable data created by the fieldbus master reaches the PFC as input variables. Data created in the PFC is sent to the fieldbus master through output variables.

In the PFC, the controller can access the MODBUS TCP PFC variable data from word address 256 to 511 (double word address 128-255, byte address 512-1023) and the Ethernet IP PFC variable data from word address 1276 to 1531 (double word address 638-765, byte address 2552-3063).

3.1.6.6.1 Example MODBUS/TCP Master and PLC functionality (CPU)

Data Access by the MODBUS/TCP Master

With MODBUS TCP, the fieldbus master can access controller data as words or bits.
When accessing the first 256 words of memory from the Fieldbus port (physical I/O modules), the I/O modules start with the address 0 for both bit and word access.
When accessing data from the starting memory address of 256, the bit and word addresses follow the sequence below:
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 calculated from the following formula:

BitNo = (Word * 16) + Bitno_in_Word

Data Access by the PLC Functionality

When accessing the same data from both a PLC and Fieldbus master, the following memory address conventions should be understood:

A 16 bit IEC 61131-3 variable uses the same addressing as the MODBUS word format.

An IEC 61131-3 boolean variable (1 bit) uses a “WORD.BIT” notation for addressing, which is different from MODBUS’s bit notation.
The “WORD.BIT” notation is composed of the boolean’s word address and bit number in the word, separated by a dot. The Word and Bit values are zero based (e.g., %IX0.0 is the first possible digital input).

Example:
MODBUS bit number 19 => bit addressing in PLC <Wordno>.<Bitno> = 1.2

The PLC functionality of the PFC can also access the data as Bytes and Double- Words.

The byte addresses are computed according to the following formula:

High-Byte Address = Word address*2
Low-Byte Address = (Word address*2) + 1

The double word address is computed according to the following formula:

Double word address = High word address/2 (rounded off)
or = Low word address/2

More information
You can find a detailed description of the MODBUS addressing and the correspondent IEC61131-addressing in the chapter „MODBUS Register Mapping“.

3.1.6.6.1.1 Example of Use:

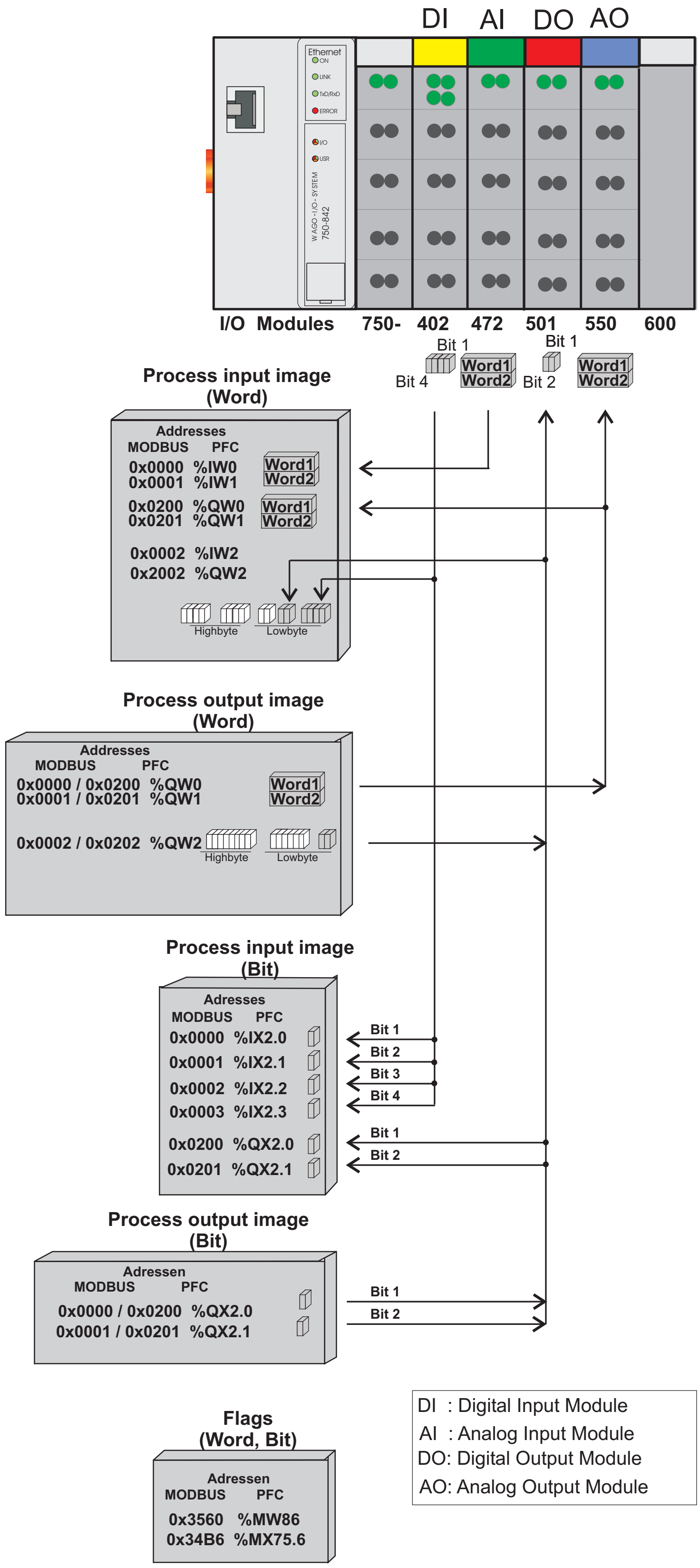


Fig. 3.1-3: Example: Addressing of a Fieldbus node g012948e

3.1.7 Starting up a Fieldbus Node

This chapter shows a step-by-step procedure for starting up a WAGO ETHERNET TCP/IP fieldbus node.
A prerequisite for communication with the controller is the assignment of an IP address. For this two different variations are described:

- **Variation 1:** Start up with the WAGO Ethernet Settings
  (offers a comfortable fast IP address assignment over the serial Configuration interface of the controller)
- **Variation 2:** Start up with the WAGO BootP Server
  (IP address assignment over the field bus, whereby compared with Variantion 1 several steps are necessary).

Additionally in the following chapters, it covers details regarding PFC programming with WAGO-I/O-*PRO* CAA and provides information about the built-in HTML web pages.

3.1.7.1 Variation 1: Start up with the WAGO Ethernet Settings

This procedure contains the following steps:

1. Connecting the PC and fieldbus node
2. Allocation of the IP address to the fieldbus node
3. Function of the fieldbus tests

3.1.7.1.1 Connecting PC and Fieldbus Node

Connect the assembled ETHERNET TCP/IP fieldbus node on your PC with the communication cable (Item-No. 750-920) between the controller's configuration and programming interface and a free serial PC port.

Warning
The communication cable 750-920 must not be connected or disconnected while the coupler/controller is powered on!

Once the operating voltage has been switched on, the PFC initialization starts. The fieldbus controller determines the configuration of the I/O modules and creates the process image. During the startup the 'I/O' LED (Red) flashes at a high frequency.

When the I/O LED turns green, the fieldbus controller is ready for operation. If an error has occurred during startup, a fault code is flashed on the 'I/O'-LED. If the I/O LED flashes 6 times (indicating error code 6) and then 4 times (indicating error argument 4), an IP address has not been assigned yet.

3.1.7.1.2 Allocating the IP Address to the Fieldbus Node

The following describes how to allocate the IP address for a fieldbus node using the WAGO Ethernet Settings by way of an example.

Note
You can download a free copy of the WAGO Ethernet Settings" which you can find on the „ELECTRONICC Tools and Docs“ CD ROM (Item-No.: 0888-0412-0001-0101) and on the WAGO Web pages under www.wago.com, "Service → Downloads → Software".
For a short description to this you can find the "Quick Start" for the ETHERNET Fieldbus Controller 750-841. Please have a look on the WAGO Web pages under www.wago.com, "Service → Downloads → Documentation".

4. Start the programm "WAGO Ethernet Settings".
5. Chose the register "TCP/IP".
6. In order to give the address now, change the specified option for the address assignment.
   By default, the address is assigned automaticly with the BootP Server. Activate now the option "Using following address" by clicking on the radio button before this option.
7. Enter the desired IP address and if necessary the address for the Subnet Mask and for the Gateway.
8. Click on the button "Write", to write the address down on the controller.

3.1.7.1.3 Testing the Function of the Fieldbus Node

1. To test the controller’s newly assigned I/P address, start a DOS window by clicking on the ***Start*** menu item ***Programs/MS-DOS Prompt***
2. In the DOS window, enter the command: "**ping** " followed by the PFC’s IP address in the following format:
   ping [space] XXX . XXX . XXX . XXX (=IP address).
   Example: ping 10.1.254.202

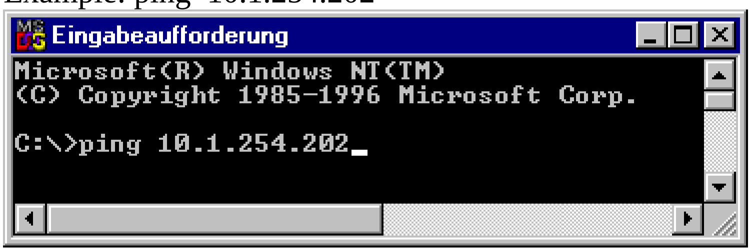


Fig. 3.1-11: Example for the Function test of a Fieldbus Node P012910d

3. When the Enter key has been pressed, your PC will receive a query from the controller, which will then be displayed in the DOS window.
   If the error message: "Timeout" appears, please compare your entries again to the allocated IP address and check all connections. Verify that the TXD/RXD LEDs flash when the ping command is issued.

4. When the test has been performed successfully, you can close the DOS prompt.
5. Since the IP address is still **temporarily** stored in the controller. Do not cycle power on the controller until the BootP protocol has been disabled in the PFC.

3.1.7.2 Variation 2:Starting up with the WAGO BootP Server

This procedure contains the following steps:

1. Noting the MAC-ID and establishing the fieldbus node
2. Connecting the PC and fieldbus node
3. Determining the IP address
4. Allocation of the IP address to the fieldbus node
5. Function of the fieldbus tests
6. Deactivating the BootP Protocol

Note
When starting up the 750-841 controller, there are a number of important factors to consider, since the start-up of this controller differs significantly in certain respects from the 750-842 ETHERNET controller.

3.1.7.2.1 Note the MAC-ID and establish the Fieldbus Node

Before establishing your fieldbus node, please note the hardware address (MAC-ID) of your ETHERNET fieldbus controller.
This is located on the rear of the fieldbus controller and on a self-adhesive tear-off label on the side of the fieldbus controller.

MAC-ID of the fieldbus controller: ----- ----- ----- ----- ----- -----.

3.1.7.2.2 Connecting PC and Fieldbus Node

Connect the assembled ETHERNET TCP/IP fieldbus node to a hub using a standard Ethernet cable, or directly to the PC with a “crossover” cable. The transmission rate of the controller is dependant on the baud rate of the PC network interface card.

Attention
For a direct connection to a PC, a “crossover” cable is required instead of a parallel cable.

Now start the BootP server on the PC and apply power to the controller (DC 24 V power pack). Once the operating voltage has been switched on, the PFC initialization starts. The fieldbus controller determines the configuration of the

I/O modules and creates the process image. During the startup the 'I/O' LED (Red) flashes at a high frequency.

When the I/O LED turns green, the fieldbus controller is ready for operation. If an error has occurred during startup, a fault code is flashed on the 'I/O'-LED. If the I/O LED flashes 6 times (indicating error code 6) and then 4 times (indicating error argument 4), an IP address has not been assigned yet.

3.1.7.2.3 Determining IP Addresses

If your PC is already connected to an ETHERNET network, it is very easy to determine the IP address of your PC. To do this, proceed as follows:

1. Go to the **Start menu** on your screen, then go to the menu item **Settings/Control Panel** and click on **Control Panel.**

2. Double click the icon **Network**  Network
   The network dialog window will open.
3. - Under <u>Windows NT</u>: Select the tab: **Protocols** and highlight the listbox item *TCP/IP protocol.*
   - Under <u>Windows 9x</u>*:* Select the tab: **Configuration** and highlight the listbox item *TCP/IP network card.*".

Attention
If the entry is missing, please install the respective TCP/IP component and restart your PC. The Windows-NT installation CD, or the installations CD for Windows 9x is required for the installation.

4. Then, click the **Properties** button. The IP address and the subnet mask are found in the **IP address** tab. If applicable, the gateway address of your PC is found in the **Gateway** tab.
5. Please write down the values:
   IP address PC: ----- . ----- . ----- . -----
   Subnet mask: ----- . ----- . ----- . -----
   Gateway: ----- . ----- . ----- . -----
6. Now select a desired IP address for your fieldbus node.

Attention
When selecting your IP address, ensure that it is in the same local network in which your PC is located.

7. Please note the IP address you have chosen:
   IP address fieldbus node: ----- . ----- . ----- . -----

3.1.7.2.4 Allocating the IP Address to the Fieldbus Node

A prerequisite for communication with the controller is the assignment of an IP address.
The address can be transferred through the "WAGO BootP Server" or a PFC

program.
With the PFC program, this is possible in **WAGO-I/O-*PRO* CAA** using the fuction block "ETHERNET_Set_Network_Config" of the library „Ethernet.lib“.

The following describes how to allocate the IP address for a fieldbus node using the WAGO BootP server by way of an example.

You can download a free copy of the WAGO’s BootP server over the Internet under at http://www.wago.com/-
“Service“/“Downloads“/“Software“/“ELECTRONICC“/-
‘WAGO BootPServer V1.0 Windows 95/NT - ZIP Archiv‘.

Note
The IP address can be allocated under other operating systems (e.g. under Linux) as well as with any other BootP servers.

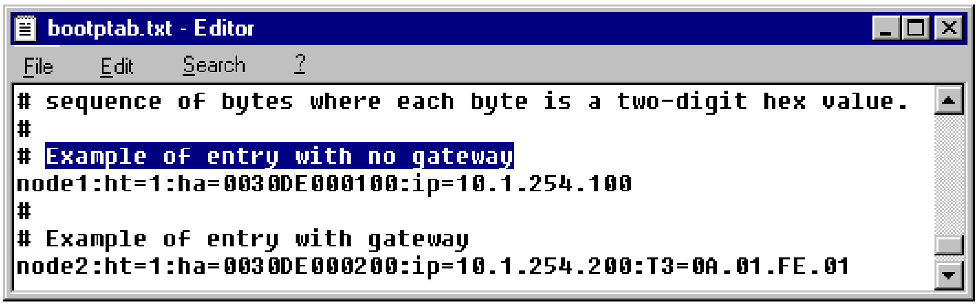
Attention
The IP address can be allocated in a direct connection via a crossover cable or via a parallel cable and a hub. An allocation over a switch is not possible.

BootP table

Note
A prerequisite for the following steps is the correct installation of the WAGO BootP server.

1. To start the BootP server, click on the ***Start*** menu item ***Programs/WAGO Software/WAGO BootP Server***.

2. After the BootP Server is started, click on the **Edit Bootptab** button located on the right hand side of the display. An editable file will appear in Windows NotePad (bootptab.txt). This file is a database for the BootP server. The file contains two examples for the allocation of an IP address, the example commands are directly after the following comment lines:
   - "Example of entry with no gateway"
   - "Example of entry with gateway"


bootptab.txt - Editor

File Edit Search ?

```
# sequence of bytes where each byte is a two-digit hex value.
#
# Example of entry with no gateway
node1:ht=1:ha=0030DE000100:ip=10.1.254.100
#
# Example of entry with gateway
node2:ht=1:ha=0030DE000200:ip=10.1.254.200:T3=0A.01.FE.01
```

Fig. 3.1-12: BootP table p012908d

The examples mentioned above contain the following information:

Declaration	Meaning
node1, node2	Any name can be given for the node here.
ht=1	Specify the hardware type of the network here. The hardware type for ETHERNET is 1.
ha=0030DE000100 ha=0030DE000200	Specify the hardware address or the MAC-ID of the ETHERNET fieldbus controller (hexadecimal).
ip= 10.1.254.100 ip= 10.1.254.200	Enter the IP address of the ETHERNET fieldbus controller (decimal) here.
T3=0A.01.FE.01	Specify the gateway IP address here. Write the address in hexadecimal form.
Sm=255.255.0.0	In addition enter the Subnet-mask of the subnet (decimal), where the ETHERNET fieldbus controller belongs to.

No gateway is required for the local network described in this example. Therefore, the first example: **"Example of entry with no gateway"** can be used.

3. Cursor to the text line: "node1:ht=1:ha=0030DE000100:ip=10.1.254.100" and replace the 12 character hardware address, which is entered after “ha=”, with your PFC’s MAC-ID.
4. If you want to give your fieldbus node a different name, replace the name "node1" with your new name.
5. To assign the controller an IP address, replace the IP address specified in the example, which is entered immediately after “ip=”, with the IP address you have selected. Make sure you separate the 3 digit numbers with a decimal point.
6. Because the second example is not necessary in this exercise, insert a “#” in front of the text line of the second example: "# hamburg:hat=1:ha=003 0DE 0002 00:ip=10.1.254.200:T3=0A.01.FE.01", so that this line will be ignored.

Note
To address more than one fieldbus nodes, add a line of setup information for each additional PFC in the file bootptab.txt . Use steps 2 through 4 as a guideline for configuring each additional module.

7. Save the new settings in the text file "bootptab.txt". To do this, go to the ***File*** menu, menu item ***Save,*** and then close the editor.

BootP Server

8. Now open the dialog window for the WAGO BootP server by going to the **Start menu** on your screen surface, menu item **Program / WAGO Software / WAGO BootP Server** and click on **WAGO BootP Server.**
9. After the editor closes, Click on the **Start** button in the opened BootP dialog window. This will activate the inquiry/response mechanism of the BootP

protocol. A series of messages will be displayed in the BootP server message window. The error messages indicate that some services (e.g. port 67, port 68) in the operating system have not been defined. DO NOT BE ALARMED, THIS IS THE CORRECT OPERATION FOR THIS EXAMPLE.

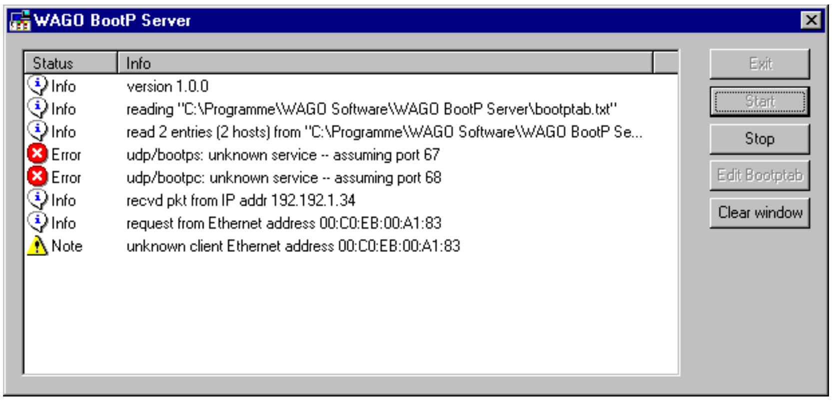


Fig. 3.1-13 Dialog Window of the WAGO BootP Server with Messages P012909d

10. Now it is important to restart the controller by resetting the hardware. To do this, cycle power to the fieldbus controller for approximately 2 seconds or press the operating mode switch down, which is located behind the configuration interface flap on the front of the controller.
    Following this, you should see a reply from the PFC stating that the IP address has been accepted (no errors). The IP address is now **temporarily** stored in the controller. Do not cycle power on the controller until the BootP protocol has been disabled in the PFC.

11. Click on the **Stop** button, and then on the **Exit** button to close the BootP Server .

3.1.7.2.5 Testing the Function of the Fieldbus Node

1. To test the controller’s newly assigned I/P address, start a DOS window by clicking on the ***Start*** menu item ***Programs/MS-DOS Prompt***

2. In the DOS window, enter the command: "**ping** " followed by the PFC’s IP address in the following format:
   ping [space] XXXX **.** XXXX **.** XXXX **.** XXXX (=IP address).
   Example: ping 10.1.254.202

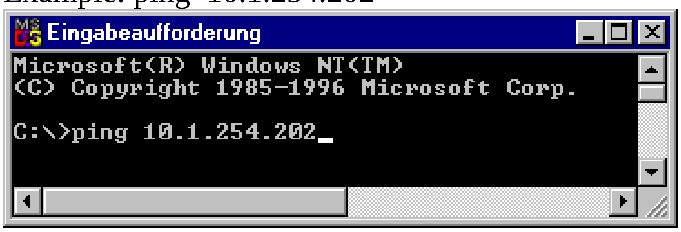


Fig. 3.1-14: Example for the Function test of a Fieldbus Node P012910d

3. When the Enter key has been pressed, your PC will receive a query from the controller, which will then be displayed in the DOS window.
   If the error message: "Timeout" appears, please compare your entries again to the allocated IP address and check all connections. Verify that the TXD/RXD LEDs flash when the ping command is issued.

4. When the test has been performed successfully, you can close the DOS prompt.
5. Since the IP address is still **temporarily** stored in the controller. Do not cycle power on the controller until the BootP protocol has been disabled in the PFC.

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 permanent presence of a BootP server. If, however, there is no BootP server available at a power-on reset, the PFC's network remains inactive.

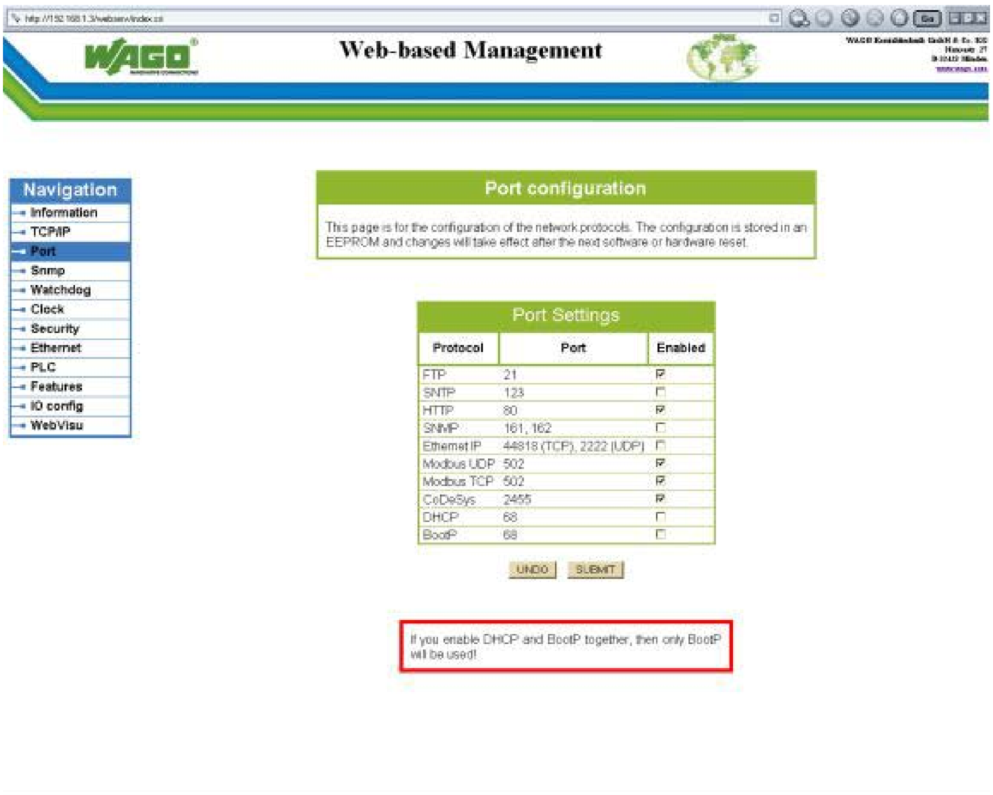
To operate the controller with the IP configuration stored in the EEPROM, the BootP protocol must be deactivated.

Note
If the BootP protocol is disabled after the IP address assignment, the stored IP address is retained even after power is removed from your controller.

1. Disabling of the BootP protocol is done via the built-in web pages stored in the controller. Open a web browser on your PC (e.g., Microsoft Internet Explorer).
2. Now enter the controller's I/P address in the address box of the browser and press the **Enter** key.
3. One of the controller's built-in web pages is displayed. The opening page displays information about your fieldbus controller. Click the **Port** hyperlink on the left navigation bar.

4. A dialog window will open and ask for a password. This serves as access protection, and includes the three different user groups: admin, guest and user.
5. To logon as the administrator, enter the user name **admin** and the password **wago**.

Note
If the controller does not display the opening HTML page, make sure your web browser is setup to bypass the proxy server for local addresses.



6. A list of all protocols supported by the controller is displayed. The BootP protocol is activated by default. To disable the protocol, click on the check box after **BootP** to remove the check mark.
7. You can disable other protocols you do not need in a similar way, or enable protocols you wish to use. It is possible to enable several protocols at the same time, since each protocol uses a different port.
8. To store the protocol selection, click the **SUBMIT** button and then perform a hardware reset. To do this, either switch off the power supply of the controller or press down the operating mode switch.
9. The protocol settings are now stored EEPROM and the controller is ready to operate.

3.1.7.3 Transmission Mode Configuration

Both the controller and its link partner must be configured for the same transmission mode to ensure reliable and fast communication using the ETHERNET TCP/IP controller, which means they must operate either in (default) autonegotiation mode or full or half duplex mode with 10/100Mbit static transmission rate.

If the transmission mode must be set to match the link partner, the "Ethernet" link can be used to access a web page on which the transmission rate and the bandwidth limit can be adjusted for the Ethernet transmission. These values should only be modified in exceptional circumstances.

Note
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.

Further Information
Please find detailed information on how to configure the transmission mode in the chapter "Fieldbus Communication"/"ETHERNET"/
"Fieldbus Communication"/"ETHERNET"/
"Network Architecture – Principles and Regulations"/"Transmission Mode".

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

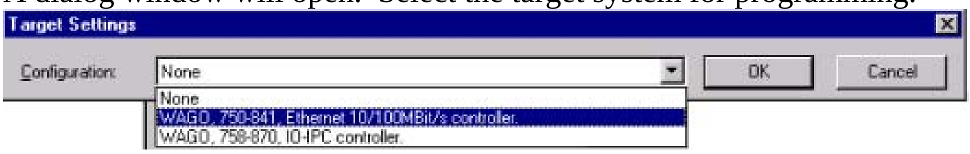
The WAGO 750-841 Programmable Fieldbus Controller (PFC) combines the functionality of an ETHERNET fieldbus coupler with the functionality of a Programmable Logic Controller (PLC). When the PFC is used as a PLC, all or some of its I/O modules can be control locally with the use of WAGO-I/O-*PRO* CAA. WAGO-I/O-PRO CAA is an IEC 61131-3 programming tool that is used to program and configure the 750-841 PFC. I/O modules which are not controlled locally (i.e., not controlled as a PLC), can be controlled remotely through the 10/100 Mbps ETHERNET Fieldbus port.

Note
To perform IEC 61131-3 programming in the 750-841 PFC, the WAGO-I/O-PRO port must be enabled. Enable and Disabling of this port is done with a checkbox in the “Port configuration” web page.

The purpose of this section is not to provide a comprehensive lesson on WAGO-I/O-*PRO* CAA programming. Instead, it highlights important programming and configuration notes of the IEC 61131-3 program when it is used with the 750-841 PFC.

More information
For a detailed description of how to use the software, please refer to the WAGO-I/O-*PRO* CAA manual. An electronic copy of this manual can be found on WAGO’s web site: **www.wago.com**

1. To start the WAGO-I/O-*PRO* CAA, click the ***Start*** menu item ***Programs/CoDeSys for Automation Alliance/CoDeSys V2.3/ CoDeSys V2.3*** .
   A dialog window will open. Select the target system for programming.



2. Choose **WAGO, 750-841, Ethernet 10/100Mbps controller** from the pull down list and click the **OK** button.

3. You can now create a new project in WAGO-I/O-PRO CAA via its menu item ***File/New***. A dialog window will prompt you to select the programming language (i.e., IL, LD, FBD, SFC, etc.).

4. To access I/O modules of your node, the module configuration must first be mapped in the file "EA-config.xml". This file defines which system may have write access to each particular I/O module (i.e., the IEC 61131-3 program, MODBUS TCP Fieldbus, or Ethernet IP Fieldbus).
   This file can be generated, as described in the following, by the configuration with the WAGO I/O *PRO* CAA Configurator.

Configuration with the WAGO-I/O-PRO CAA I/O-Configurator

1. To configure the I/O of the fieldbus node in WAGO-I/O-PRO, select the **Resources** tab in the left window of the screen, then double click on **PLC configuration** in the tree structure.
   The I/O-Configurator is starting.

2. Expand the branch '**Hardware configuration**' and the sub-branch **'K-Bus'** in the tree structure.

3. Click the entry '**K-Bu**s' or an I/O module with the right mouse key to open a context menu to insert and append of I/O modules.

4. If the K-Bus structure is empty or the entry '**K-Bu**s' has been selected, use the command '**Append Subelement**' to select the desired I/O module and attach it to the end of the K-Bus structure. The command 'Insert Element' is inactive in this case.

5. If an I/O module has been selected in the K-Bus structure, use the command **'Insert Element**' to select the desired I/O module and insert it into the structure above the selected position. The command '**Append Subelement**' is inactive in this case.

6. The corresponding commands are also accessible via the menu **'Insert**' in the menubar of the main window.

7. Both commands open the dialog box **'I/O-configuration**'.

8. In this dialog box you can select a desired I/O module from the catalog and place it in the node configuration. Place all necessary I/O modules in the node configuration until the configuration fits to the physical node.
   Add a module to the tree structure for each module in your node that supplies or expects data in bits or words.

Attention
The number of modules that you add must agree with the physical hardware present (except for supply modules, potential multiplication modules, and end modules).

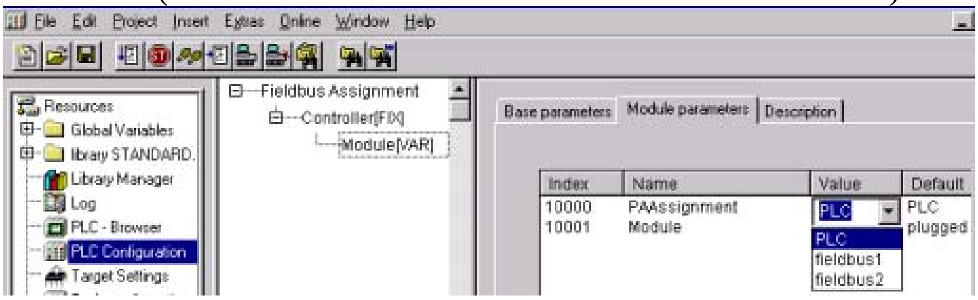
9. To get more information about an I/O module, select the desired module either from the catalog or from the node configuration and press the '**Data Sheet**' button. The data sheet corresponding to the selected module will be opened in a new window.

Note
You will find the most current version of the data sheets in the internet under www.wago.com.

10. Accept changes in the node configuration and close the dialog box by pressing the '**OK**' button. The addresses in the PLC-configuration are recalculated and the tree structure is refreshed afterwards.

11. You can change now the access authorization, if for individual modules, the write access should be via the fieldbus (MODBUS TCP/IP or Ethernet/IP). For each module added, the write access first is fixed from the PLC. You can change it by determining in the right-hand dialogue window/register "**Module parameter**s for each individual module, from where the write access on the I/O Module data should take place.

    For this in the "**Value**" column. your options include:
    - PLC (The PFC controls its I/O locally)
    - fieldbus 1 (A MODBUS TCP Fieldbus Master controls the I/O module)
    - fieldbus 2 (An Ethernet IP Fieldbus Master controls the I/O module)



12. When you have completed the assignments, you can start programming with the IEC 61131-3 program tool. The configuration file “EA-config.xml” is generated as soon as you compile the project.

More information
For a detailed description on how to use the software WAGO-I/O-PRO CAA and the I/O-Configurator, please refer to the Online-Help of WAGO-I/O-*PRO* CAA.

Note
Alternatively, the "EA-config.xml" file can be created with each editor and then be transported via FTP in the Controller directory "/etc".
The configuration with the “EA-config.xml” file, which is already stored in the controller, is described in the following chapter.

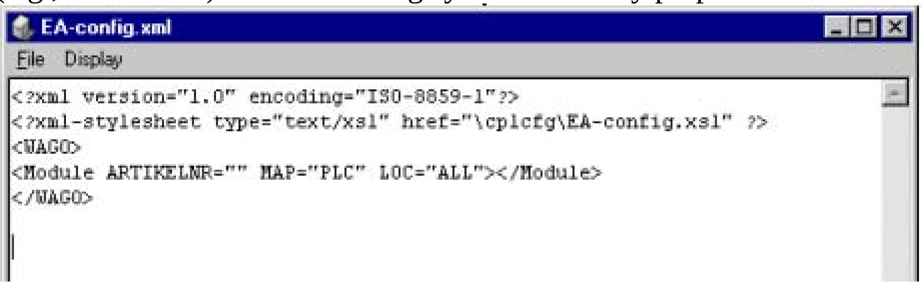
Configuration with the “EA-config.xml” file

Note
If you wish to directly assign the module mapping using the EA-config.xml file stored in the controller, you must not have previously stored any configuration settings in WAGO-I/O PRO, since this file will be overwritten by the entries in WAGO-I/O-PRO on performing a download

1. Open the FTP client you wish to use (e.g., “LeechFTP”, which is freely downloadable on the Internet).

2. To access the file system of the controller, enter the IP address of the controller in the FTP client. Also, set the user name to **admin** and the password to **wago**. The “EA-config.xml” file can be found in the folder *etc.* on the PFC server.

3. Copy the file into a local directory on your PC and open it with a text editor (e.g., “NotePad”). The following syntax is already prepared in the file:



4. The fourth line contains the necessary information for the first module. The entry [MAP=“PLC“] assigns control rights to the IEC 61131-3 program for the first module. If you want to change the control setting, replace “PLC” with “FB1” for control rights from MODBUS TCP, or with “FB2” for control from Ethernet IP.

5. Now add under the fourth line the same syntax for each individual module with the appropriate control assignment:
<Module ITEM NO.=““ MAP=“(e.g.) PLC” LOC=“ALL”></Module>.

Note
The number of line entries must agree with the number of hardware modules present in your node.

6. Save the file and download it back to the file system of the controller using the FTP client.

7. You can now start programming with WAGO-I/O-*PRO* CAA.

More information
For a detailed description on how to use the software, please refer to the WAGO-I/O-*PRO* CAA manual. An electronic copy of this manual can be found on WAGO’s web site: www.wago.com

3.1.8.1 WAGO-I/O-*PRO* CAA library elements for ETHERNET

WAGO-I/O-*PRO* CAA offers various libraries for different IEC 61131-3 programming applications. They contain a host of modules that facilitate and speed up the creation of your application program.

More Information
You can find all libraries on the WAGO-I/O-*PRO* CAA installation CD ROM in the folder: **CoDeSys V2.3\Targets\WAGO\Libraries\...**

Some libraries, e. g. the 'standard.lib', are by default included in a new project. The table below describes some of the other libraries that are particularly available for ETHERNET projects with WAGO-I/O-*PRO* CAA.

Ethernet. lib	provides function blocks for communication over ETHERNET
WAGOLibEthernet_01.lib	provides function blocks, which allow to establish a connection to a remote server or client using TCP protocol and to use UDP protocol to exchange data with any UDP server or client.
WAGOLibModbus_IP_01.lib	provides function blocks for establishing communication with one or more slaves
ModbusEthernet_03.lib	provides function blocks for establishing communication with one or more Modbus slaves
ModbusEthernet_04.lib	provides function blocks for the data exchange with several Modbus TCP/UDP Slaves and in addition one function block, which makes a Modbus server available and illustrates the Modbus services on an word array
SysLibSockets.lib	allows the access on sockets for communication over TCP/IP and UDP
WagoLibSockets.lib	allows the access on sockets for communication over TCP/IP and UDP and provides additional functions to SysyLibSockets.lib
WAGOLibMail_01.lib	provides function blocks for sending E-Mails:
Mail_02.lib	allows sending Emails
WagoLibSnmpEx_01.lib	allows sending SNMP-V1-Trap’s together with parameters of type DWORD and STRING(120) (starting from software version SW >= 07).
WagoLibSntp.lib	provides function blocks for the settings and the use of the Simple Network Time Protocol (SNTP)
WagoLibFtp.lib	provides function blocks for the settings and the use of the File Transfer Protocol (FTP)

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

After installing these libraries, you will have access to their POUs (Program Organization Units), data types, and global variables, which can be used in the same manner as user defined program objects.

More information
For information on the function blocks as well as details regarding the use of the software, please refer to the WAGO-I/O-*PRO* CAA manual or the onlinehelp. An electronic copy of the manual can be found on WAGO’s web site: **www.wago.com**.

3.1.8.2 Restrictions in the Function Range

The basic of WAGO-I/O-*PRO* CAA, the standard programming system CoDeSys from the company 3S, has three process variants "HMI", "TargetVisu" and "WebVisu"within the integrated visualisation system.

The 750-841 Ethernet controller supports the process variants "HMI" and "WebVisu". There are technological limitations depending on the process variant.

Different options of the complex visualisation objects "Alarm" and "Trend" are available exclusively in the "HMI". This applies, for example, for sending e-mails as a response to an alarm or for navigating through and generating historical trend data.

On the Ethernet Controller 750-841 the "WebVisu" is executed within considerably tighter physical limits, compared with "HMI". Whereas the "HMI" can call upon the almost unrestricted resources of a PC, the "WebVisu" must take into account the following restrictions:

Technological Limitations of the CoDeSys WebVisu	
File System (1,4 MB)	The total size of PLC program, visualisation files, bitmaps, log files, configuration files etc. must fit into the file system. The PLC browser returns the free memory available with the command "fds" (FreeDiscSpace).
Process Data Buffer (16 kB)	The WebVisu uses its own protocol for exchanging process data between applet and control system. In doing so, the process data is transmitted with ASCII coding. The pipe character ("|") is used as a separator between two process values. The space required for a process data variable in the process data buffer is therefore not only dependent on the type of data but also on the process value itself. A "WORD" variable therefore occupies between one byte for the values 0... 9 and five bytes for values above 10000. The selected format allows only a rough estimate of the space required for the individual process data in the process data buffer. If the size is exceeded, the WebVisu no longer works as expected.
Number of Modules (512/default)	The total size of the PLC program is determined, amongst other things, by the maximum number of modules. This value can be configured in the target system settings.
Computer Power/Processor Time	is based on a real-time operating system with pre-emptive multitasking. With this system, high-priority processes, such as the PLC program, for example, interrupt or suppress low-priority processes. The web server, which is responsible for supplying the

	applet and exchanging process data with the applet, is such a low-priority process. Make sure when configuring tasks that there is sufficient processor time available for all processes. The "freewheeling" task call option is not suitable in conjunction with the "WebVisu", as in this case the high-priority PLC program suppresses the web server. Instead of this, use the "cyclic" task call option with a realistic value. The PLC browser provides an overview of the real execution times of all CoDeSys tasks with the command "tsk". If operating system functions for the handling of "sockets" or the "file system" are used in a PLC program for example, these execution times are not taken into account by the "tsk" command.
Network Loading	The Ethernet controller 750-841 has just one CPU, which is responsible both for running the PLC program and for handling the network traffic. Ethernet communication demands that every telegram received is processed, regardless of whether it is intended for the 750-841 or not. A significant reduction in network loading can be achieved by using "switches" instead of "hubs". However, broadcast telegrams can only be suppressed by the sender or by means of configurable switches, which feature broadcast limiting. A network monitor such as www.ethereal.com will give an overview of your current network loading.
	Attention Please pay attention that the bandwidth limit, which can be configured in the Web Based Management System under the „Ethernet“ link. It is not a suitable means for increasing the operating reliability of the "WebVisu", as in this case telegrams are ignored or rejected.

More Information
It is not possible to define hard key data, because of the reasons above.
So please take as support for your planning the Application Notes, published in the InterNet, with appropriate projects showing the efficiency of the Web visualization.
Please find Application Notes under www.wago.com.

3.1.8.3 Some Basic Facts about IEC Tasks

Attention
Consider please with the programming of your IEC tasks the following facts.

- All IEC tasks must have a different priority level. If two tasks have the same level, an error message is displayed when the program is compiled.
- A running task can be interrupted by a task of higher priority. The interrupted task will resume execution after all higher priority tasks are completed.
- If several IEC tasks use the same input/output variables from the process image, the values of the input /output variables can change during the execution of each IEC task, causing contention in the application program.
- Freewheeling tasks pauses after each task cycle for 1ms, before the renewed execution begins.
- If no task is applied in the task configuration, a freewheeling default task is applied when the program is compiled. The watchdog of this task is deactivated. This task, named "DefaultTask", is internaly identified with this name, so don’t use this name for your own tasks.
- Only for cyclic tasks the sensitivity value is used. Sensitivity values of 0 and 1 are synonymic. A value of 0 or 1 means that the watchdog triggers on a single cycle time overrun. With a sensitivity value of 2 for example, the watchdog triggers on two consecutive cycle time overruns.
- To cyclic tasks with activated watchdog applies:
  - Is the adjusted maximum runtime less than the sampling rate, the watchdog will also trigger if the runtime exceeds the sampling rate, irrespective of the value that has been entered for the sensitivity.
  - Is the adjusted maximum runtime greater then the sampling rate, the watchdog will trigger if the maximum runtime is exceeded, irrespective of the value that has been entered for the sensitivity.

3.1.8.3.1 Flowchart of an IEC Task

- Get system time (tStart).
- If the last I/O bus cycle is not complete.
  -> Wait for the end of the next I/O bus cycle.
- Read the inputs and the outputs from the process image.
- If the user application program is running.
  -> Execute the program code of this task.
- Write the outputs into the process image.
- Get system time (tEnd).
  -> tEnd - tStart = run-time of the IEC task.

3.1.8.3.2 Overview of the Most Important Task Priorities (descending priority)

- I/O Bus Task / Fieldbus Task (Internal):
  The I/O Bus task is an internal task, which updates the I/O module data from the process image.
  Fieldbus tasks are triggered by fieldbus events (communications); therefore, they only use processing time when the fieldbus is active.

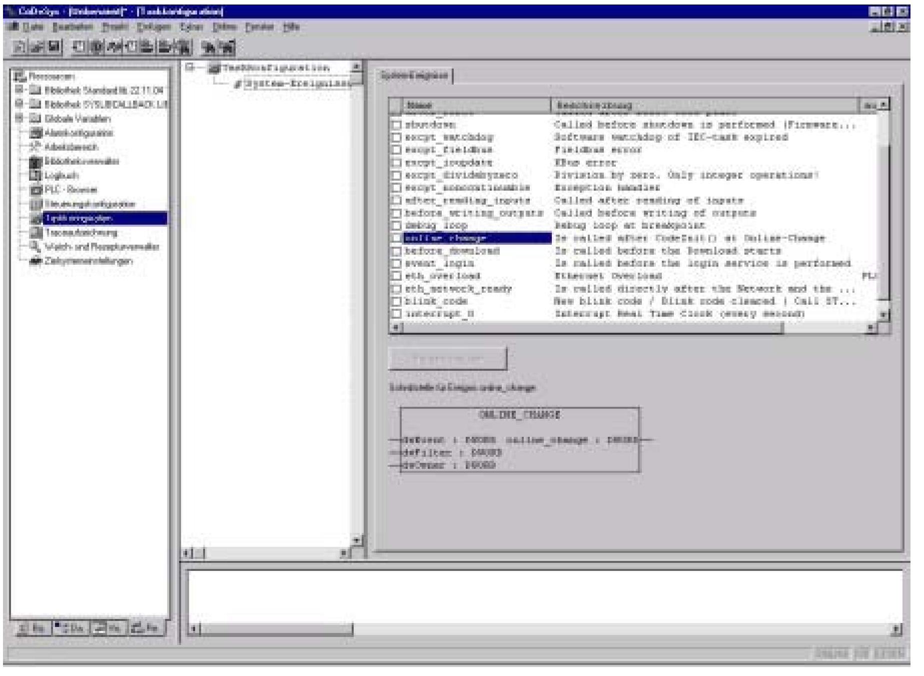
- Normal Task (IEC-Tasks 1-10):
  IEC tasks with these priorities can be interrupted by the I/O Bus and Fieldbus tasks. Therefore, if the watchdog is used, before selecting the sampling rate, consider the number of I/O modules in the node and the communication activity via the fieldbus.

- PLC Comm Task (Internal):
  The PLC Comm Task is only active when you are logged in with CoDeSys. This task manages the communication with the CoDeSys-Gateway.

- Background Task (IEC-Tasks 11-31):
  All internal tasks have a higher priority than the IEC background tasks. Therefore, IEC background tasks are used for time-consuming and time-uncritical jobs (e.g., SysLibFile functions)

More information
For detailed information on the programming tool WAGO-I/O-*PRO* CAA, please refer to the WAGO-I/O-*PRO* CAA manual. An electronic copy of this manual can be found on WAGO's web site: **www.wago.com**

3.1.8.3.3 System Events



Instead of a "task" also a "system event" can be used to call a POU of your project.

The available system events are target specific (definition in target file). The list of the standard events of the target may be extended by customer specific events. Possible events are for instance: Stop, Start, Online Change.
The complete list of all system events is specified in WAGO-I/O-*PRO* CAA /register "Resources"/"Task configuration"/"System events".
If you actually want the POU to be called by the event, activate the entry in the assignment table. Activating/deactivating is done by a mouse click on the control box.

More Information
The assignment of the system events to the POUs is described in detail in the manual to the programming tool WAGO-I/O-*PRO* CAA under:
www.wago.com ->Service->Downloads->Documentation

3.1.8.4 IEC 61131-3-Program transfer

Transferring an IEC 61131-3 application program from your PC to the PFC can be done in two different ways:

- via the serial RS232 interface
- via the fieldbus with TCP/IP.

Note
When choosing the driver, make sure the communication parameters are correctly set and match your controller.

The **Communication parameters** dialog window is displayed through the program menu item ***Online*** and then ***Communication parameters***.

1. If you choose the RS232 driver, verify that the **Communication parameters** dialog window contains the following setup data: Baudrate=19200, Parity=Even, Stop bits=1, Motorola byteorder=No.
2. If you choose the TCP/IP driver, verify that the **Communication parameters** dialog window contains the following setup data: Port=2455, Motorola byteorder=No. Additionally, verify that the entered IP address is correct.

More information
For information on the installation of the communication drivers, as well as details regarding the use of the software, please refer to the WAGO-I/O-*PRO* CAA manual. An electronic copy of this manual can be found on WAGO's web site: **www.wago.com**

3.1.8.4.1 Transmission via the Serial Interface

Use the WAGO communication cable to produce a physical connection to the serial interface. This is contained in the scope of delivery of the programming tool IEC 1131-3, order No.: 759-333/000-002, or can be purchased as an accessory under order No.: 750-920.
Connect the COMX port of your PC with the communication interface of your controller using the WAGO communication cable.

Warning
The communication cable 750-920 must not be connected or disconnected while the coupler/controller is powered on!

A communication driver is required for serial data transmission. In **WAGO-I/O-*PRO* CAA,** this driver and its parameterization are entered in the **"Communication parameters"** dialog.

1. Start WAGO-I/O-*PRO* CAA by using the Windows ***Start*** menu, find and click on the WAGO-I/O-*PRO* program name (i.e., ***CoDeSys V2.3***).
2. In the ***Online*** program menu, click ***Communication parameters***. The dialog window **Communication parameters** opens. Next, click **New** to create a new communications channel and the **Communication Parameters: New Channel** dialog window opens.
3. In the **Communication Parameters: New Channel** dialog window, you can enter a channel description in the “Name” field, then single click on **Serial (RS232)**. Click the **OK** button to close the dialog window and the **Communication Parameters** dialog window will regain focus.
4. In the center window of the dialog, the following parameters appear: Baud rate=19200, Parity=Even, Stop bits=1, Motorola byteorder=No.
   If necessary, change the entries accordingly. After all changes are entered, Click **OK**. You can now test communication with your controller.

   Note
   To access the controller, the operating mode switch of the controller must be in the center or the top position.
5. Under the program menu item ***Online,*** click ***Log-on*** to log on to the controller. (During online operations the WAGO-I/O-*PRO* CAA server is active and the ***Communication Parameters*** menu item cannot be accessed.)
6. If the controller does not contain a application program, or it contains a different program, a dialog window appears asking whether or not the new program should be loaded. Confirm with the **Yes** button.
7. As soon as the program is loaded, you can start the controller by clicking on the program menu item ***Online*** and then ***Run***. At the right-hand side of the status bar (the status bar is located at the bottom of the screen), the **RUNNING** indicator will be highlighted .
8. To terminate the online operation, return to the ***Online*** menu and click on ***Log-off***.

3.1.8.4.2 Transmission by the Fieldbus

When using the Ethernet Fieldbus port for communication with WAGO-I/O-*PRO* CAA, the PC and the controller must be connected physically via Ethernet (refer to Figure 5-2 and 5-3 for typical network connections). Additionally, the TCP/IP communication driver in WAGO-I/O-*PRO* CAA must be setup correctly and the controller must contain an IP address (refer to section 3.1.6.4 for assigning an IP address to the controller).
A communication driver is required for Ethernet transmission. In WAGO-I/O-*PRO* CAA, this driver and its parameters are entered in the **Communication Parameters** dialog.

1. Start WAGO-I/O-*PRO* CAA by using the Windows ***Start*** menu, find and click on the WAGO-I/O-*PRO* program name (i.e., CoDeSys V2.3).

2. In the ***Online*** program menu, click ***Communication parameters***. The dialog window **Communication parameters** opens. Next, click **New** to create a new communications channel and the **Communication Parameters: New Channel** dialog window opens.

3. In the **Communication Parameters: New Channel** dialog window, you can enter a channel description in the “Name” field, then single click on **Tcp/Ip.** Click the **OK** button to close the dialog window and the **Communication Parameters** dialog window will regain focus.

4. In the center window of the dialog, the following parameters appear: Address=localHost, Port=2455, Motorola byteorder=No. Replace the Address “localHost” with the IP address of your controller assigned via the BootP server. If necessary, change the other entries accordingly. After all data has been entered, Click **OK**. You can now test communication with your controller.

Note
To access the controller, it must have an IP address, and the operating mode switch of the controller must be in the center or top position.

5. Under the program menu item ***Online,*** click ***Log-on*** to log on to the controller. (During online operations the WAGO-I/O-*PRO* CAA server is active and the ***Communication Parameters*** menu item cannot be accessed.)

6. If the controller does not contain an application program, or it contains a different program, a dialog window appears asking whether or not the new program should be loaded. Confirm with the **Yes** button.

7. As soon as the program is loaded, you can start the controller by clicking on the program menu item ***Online*** and then ***Run***. At the right-hand side of the status bar (the status bar is located at the bottom of the screen), the **RUNNING** indicator will be highlighted.

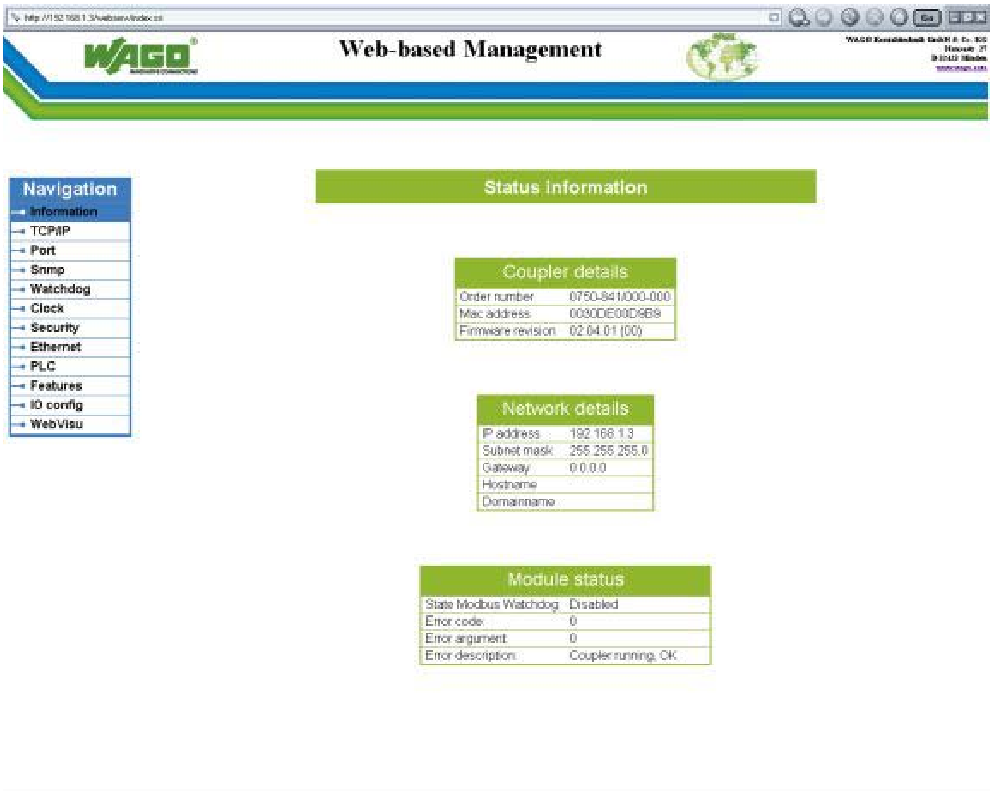
8. To terminate the online operation, return to the ***Online*** menu and click on the ***Log-off***.

3.1.9 Information on the Web-Based Management System

In addition to the web pages already described in section 3, the following HTML pages are stored in your controller and provide information and configuration options. After opening the default page of your controller, you can access these pages via the hyperlinks in the left navigation bar of the browser window.

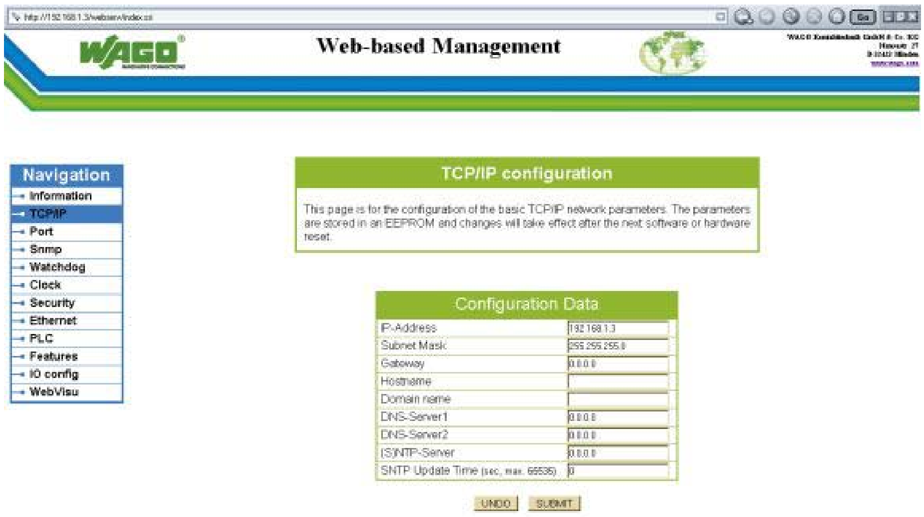
Information

Under the link **Information** you can get the status information about the controller and the network.



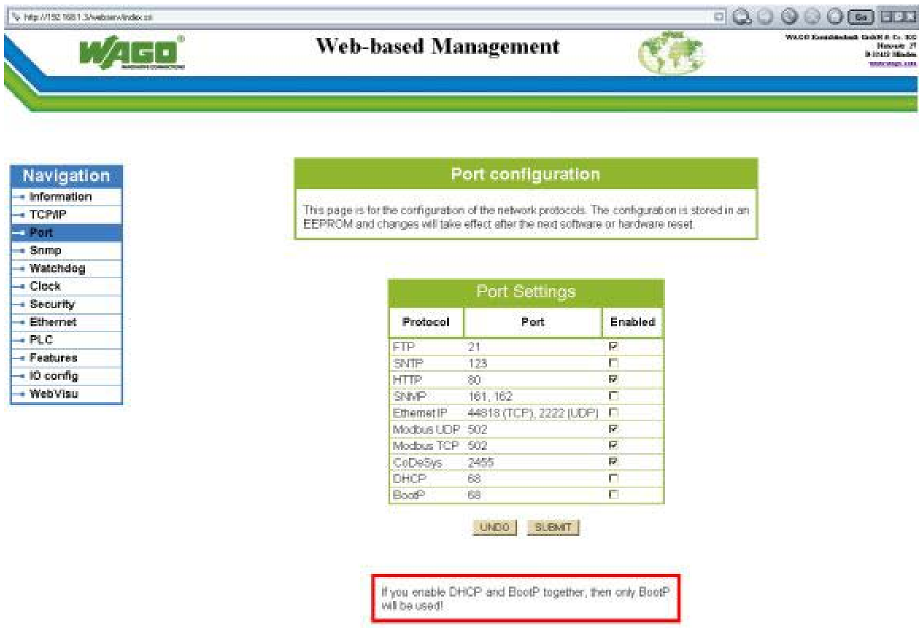
TCP/IP

Under the **TCP/IP** link, you can view and change settings for the TCP/IP protocol, which is responsible for network transmission.



Port

Under the link **Port** you can get the HTML page "Port configuration" on that you can activate or deactivate wished protocols.
FTP, HTTP, WebVisu, MODBUS TCP, MODBUS UDP, CoDeSys and BOOTP are activated by default.



Snmp

Under the **Snmp** link, you can view and change settings for the Simple Network Management Protocol, which is responsible for the transport of control data.

More information
For detailed information to the settings and the configuration of SNMP please refer the following chapter "Configuration of SNMP".

Watchdog

Under the **Watchdog** link, you can view and change settings for the MODBUS Watchdog.

Clock

Under the **Clock** link, you can view and change settings for the controller' s internal real time clock.

By the Configuration of the SNTP client the syncronization of the time of day is made. 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.

Security

Under the **Security** link, you can setup read/write access rights by using passwords for different user groups in order to protect against configuration changes.
The following groups are provided for this:

User	Pass word	Rights
admin	wago	Read, Write data and the access on the security settings
user	user	Read, Write data and the release of a Software reset, but not the change of the security settings
guest	guest	only Read

Attention
For the password the following restrictions must be considered: max. 16 signs, only letters and numbers, no special characters and umlauts.

Ethernet

Under the **Ethernet** link**,** you can view and change settings for the data transmission rate and the bandwidth limitation for the Ethernet transmission. These should be changed however only in completely special cases.

Note
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.
Both the controller and its link partner must be configured for the same transmission mode to ensure reliable and fast communication using the ETHERNET TCP/IP controller, which means they must operate either in (default) autonegotiation mode or full or half duplex mode with 10/100Mbit static transmission rate.

Further Information
Please find detailed information on how to cofigure the transmission mode in the "Fieldbus Communication"/"ETHERNET"/
"Network Architecture – Principles and Regulations "/"Transmission Mode".

The bandwidth limitation can be activated or deactivated. The "Active time (ms)" is thereby the Watchdog time, after which the interrupts become disabled, whereby telegrams can be lost. The timer is triggered by a task in the system. The priority of the task is determined by the parameter mode:

Mode 1: Only the internal bus is held by to run.
Mode 2: The internal bus and the CPU of the controller are held by to run.
Mode 3: All tasks begin to run.

PLC

Under the **PLC** link you can define the status of the outputs, if your application program stops.
If there is a checkmark in the small box behind "Enabled", all outputs are set to zero, otherwise the outputs remain to the last current value.

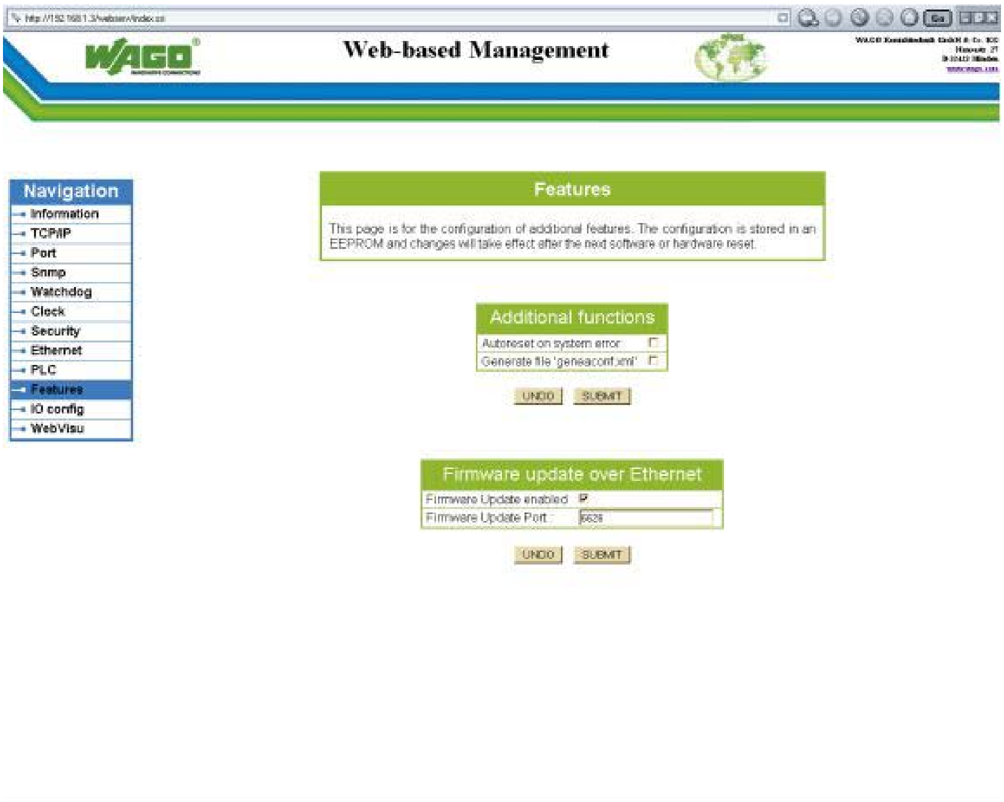
Features

Under the link **Features** you can get the HTML page "Feature" on that you can activate or deactivate additional functions.

With the function " Autoreset on system error " an automatic Software Reset in the case of an arising system error is possible.
This function can ensure a safe and durable operation with use within difficulty accessible ranges (e.g. within the offshore range). The automatic reset releases, as soon as the controller is in an error status, which requires a reset.
On delivery this function is deactivated (default), then with the arising of an error the diagnosis is signalled by the blink code of the "I/O" LED. After the evaluation of the blink code and the error removal, then a manual reset is to be accomplished.

With the function " Generate file 'genIOconf.xml' " it is possible to generate an IO configuration file, which contains beside the real fieldbus node configuration also an access mechanism on process values.
On delivery this function is deactivated (default). With the activation of this

function, the configuration file is new generated with each Power On. The file content is then represented in the right window of the web page you can get under the link "IO config".



I/O config

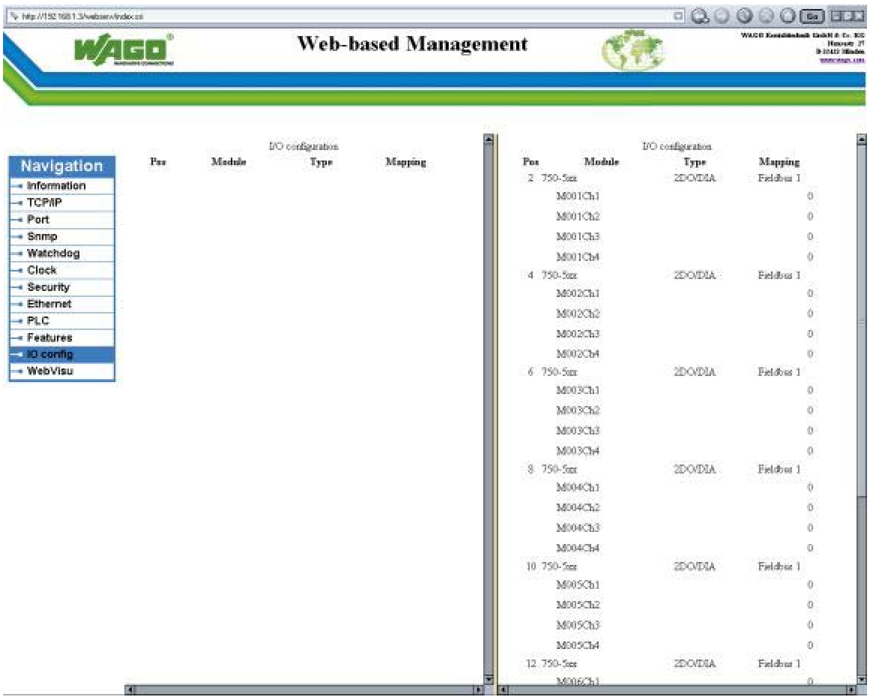
Under the link **I/O config** you can view the configuration or the write access rigths on the outputs of your fieldbus node.

In the left window of this web page is displayed the node configuration generated with the hardware configurator "**WAGO-I/O-*PRO* CAA I/O Configurator**". If no modules are indicated in this window, then no hardware configuration and also no assignment from write access rights is still taken place.
In this case, always fieldbus 1, that means Modbus_TCP (Default), has the r write access rigths on the outputs.

More Information
Please, find detailed information about the **WAGO-I/O-*PRO* CAA I/O-Configurator** in chapter 3.1.7 "Starting up a Fieldbus **Node**".

In the right window of this web page is displayed the physical node configuration and the current process values, if the file 'genIOconf.xml' was generated. A condition for the file generation is, that on the web page reached by the link "**Features** ", the function " Generate file 'genIOconf.xml' " was activated. On delivery this function is deactivated by default, so that the file

' genIOconf.xml ' is not generated and no modules and process data are represented in this window.



Samples

Under the **Samples** link, a sample HTML page is provided, which you can use as a starting point to create your own web page. You can then store this or any other web page you have created into the file system of the controller using FTP download.

3.1.10 Configuration of SNMP

The Simple Network Management Protocol (SNMP) is responsible for the transport of control data, which enables the exchange of management information, status and statistical data between individual network components and a management system. Version 1 of the protocol is supported.

SNMP represents a standard for the management of devices in a TCP/IP network. An SNMP management workstation polls the SNMP agent in order to obtain information on the corresponding devices. Components of a device, which the SNMP agent can access or an SNMP agent can modify, are described as SNMP objects. Collections of SNMP objects are contained in a logical database, the Management Information Base (MIB), and the objects are frequently referred to as MIB objects.

In the ETHERNET coupler/controller, SNMP embraces the common MIB as described in RFC1213 (MIB II).

The configuration of SNMP is done via the web-based management system under the “Snmp” link or directly via SNMP.
On the web page stored in the coupler/controller, the name of the device (sysName), the description of the device (sysDescription), the location (sysLocation) and the contact person (sysContact) can be freely set.
In addition, up to 2 trap managers can be specified.

The SNMP is executed via Port 161. The port number for SNMP traps is 162. This port number cannot be changed.

3.1.10.1 Description of MIB II

The MIB II conformance to RFC1213 is divided into the following groups:

Group	Identifier
• System Group	1.3.6.1.2.1.1
• Interface Group	1.3.6.1.2.1.2
• Address Translation Group	1.3.6.1.2.1.3
• IP Group	1.3.6.1.2.1.4
• IpRoute Table	1.3.6.1.2.1.4.21
• IpNetToMediaTable	1.3.6.1.2.1.4.22
• ICMP Group	1.3.6.1.2.1.5
• TCP Group	1.3.6.1.2.1.6
• UDP Group	1.3.6.1.2.1.7
• SNMP Group	1.3.6.1.2.1.11
• EGP Group	1.3.6.1.2.1.8

3.1.10.1.1 System Group

The System Group contains general information to the coupler/controller.

Identifier	Entry	Ac-cess	Description
1.3.6.1.2.1.1.1	sysDescr	R	This entry contains the device identification. The entry is fix coded e. g. on "WAGO 750-841".
1.3.6.1.2.1.1.2	sysObjectID	R	This entry contains the authorizing identification of the manufacturer.
1.3.6.1.2.1.1.3	sysUpTime	R	This entry contains the time in hundredth seconds since the last reset of the managements unit.
1.3.6.1.2.1.1.4	sysContakt	R/W	This entry contains the identification of the contact person and contains information about the contact possibilities.
1.3.6.1.2.1.1.5	sysName	R/W	This entry contains an administrative name for the device.
1.3.6.1.2.1.1.6	sysLocation	R/W	This entry contains the physical place of installation of the node.
1.3.6.1.2.1.1.7	sysServices	R	This entry designates the quantity of services, which this coupler/controller contains.

3.1.10.1.2 Interface Group

The interface Group contains information and statistics to the device interface.

Identifier	Entry	Ac-cess	Description
1.3.6.1.2.1.2.1	ifNumber	R	Number of network interfaces in the system
1.3.6.1.2.1.2.2	ifTable	-	Number of network interfaces
1.3.6.1.2.1.2.2.1	ifEntry	-	Entry network interface
1.3.6.1.2.1.2.2.1.1	ifIndex	R	This entry contains an unique assignment number for each interface
1.3.6.1.2.1.2.2.1.2	ifDescr	R	This entry contains the name of the manufacturer, the name of the product and the version of the hardware Interface. E. g. "WAGO Kontakttechnik GmbH 750-841: Rev 1.0"
1.3.6.1.2.1.2.2.1.3	ifType	R	This entry describes the type of the interface. Ethernet-CSMA/CD = 6, Software-Loopback = 24
1.3.6.1.2.1.2.2.1.4	ifMtu	R	This entry specified the maximum transfer unit (i.e. the maximum telegram length over this interface to be transferred can).
1.3.6.1.2.1.2.2.1.5	ifSpeed	R	This entry indicates in bits the speed of the interface.
1.3.6.1.2.1.2.2.1.6	ifPhysAddress	R	This entry indicates the physical address of the interface. In the case of Ethernet this is the MAC-ID.
1.3.6.1.2.1.2.2.1.7	ifAdminStatus	R/W	This entry indicates the desired condition of the interface. Possible values are here: up(1) : Ready for use to sending and receiving down(2) : Interface is switched off testing(3) : Interface is in the test mode

1.3.6.1.2.1.2.2.1.8	ifOperStatus	R	This entry indicates the current condition of the interface.
1.3.6.1.2.1.2.2.1.9	ifLastChange	R	This entry indicates the value of sysUpTime at the time in which the condition changed for the last time.
1.3.6.1.2.1.2.2.1.10	ifInOctets	R	This entry indicates the number of all data in bytes received via the interface.
1.3.6.1.2.1.2.2.1.11	ifInUcastPkts	R	This entry indicates the number of all received Unicast packets, which were passed on a higher layer.
1.3.6.1.2.1.2.2.1.12	ifInNUcastPkts	R	This entry indicates the number of all received Broadcast and Unicast packets, which were passed on a higher layer.
1.3.6.1.2.1.2.2.1.13	ifInDiscards	R	This entry indicates the number of all packets, which are destroyed although no disturbances are present.
1.3.6.1.2.1.2.2.1.14	ifInErrors	R	This entry indicates the number of all received incorrect packets, which were not passed on a higher layer.
1.3.6.1.2.1.2.2.1.15	ifInUnknown Protos	R	This entry indicates the number of all received packets, which were passed on an unknown or not supported port number.
1.3.6.1.2.1.2.2.1.16	ifOutOctets	R	This entry indicates the number of all data in bytes, which are transmitted so far via the interface.
1.3.6.1.2.1.2.2.1.17	ifOutUcastPkts	R	This entry indicates the number of all transmitted Unicast packets, which were passed on a higher layer.
1.3.6.1.2.1.2.2.1.18	ifOutNUcastPkts	R	This entry indicates the number of all transmitted Broadcast and Unicast packets, which were passed on a higher layer.
1.3.6.1.2.1.2.2.1.19	ifOutDiscards	R	This entry indicates the number of all packets, which are destroyed although no disturbances are present.
1.3.6.1.2.1.2.2.1.20	ifOutErrors	R	This entry indicates the number of all, which were destroyed although no disturbances are present.
1.3.6.1.2.1.2.2.1.21	ifOutQLen	R	This entry indicates the length of the queue for leaving packets.
1.3.6.1.2.1.2.2.1.22	ifSpecific	R	Always 0

3.1.10.1.3 Address Translation Group

The Address Translation Group contains information about ARP (Address Resolution Protocol) of the coupler/controller.

Identifier	Entry	Ac-cess	Description
1.3.6.1.2.1.3.1	atTable	-	Contains the allocation between network address and hardware address.
1.3.6.1.2.1.3.1.1	atEntry	-	Each entry contains the allocation between network address and hardware address.
1.3.6.1.2.1.3.1.1.1	atIfIndex	R/W	Contains the number of interface
1.3.6.1.2.1.3.1.1.2	atPhysAddress	R/W	Contains the medium independant hardware address

1.3.6.1.2.1.3.1.1.3	atNetAddress	R/W	Contains the IP address associated to the hardware address.

3.1.10.1.4 IP Group

The IP Group contains information on the IP assignment.

Identifier	Entry	Ac-cess	Description
1.3.6.1.2.1.4.1	ipForwarding	R/W	1 : Host is router; 2 : Host is not a router
1.3.6.1.2.1.4.2	ipDefaultTTL	R/W	Default value for the Time-To-Live field of each IP frame
1.3.6.1.2.1.4.3	ipInReceives	R	Number of received IP Frames including the incorrect Frames
1.3.6.1.2.1.4.4	ipInHdrErrors	R	Number of received IP Frames with header errors
.3.6.1.2.1.4.5	ipInAddrErrors	R	Number of received IP Frames with misdirected IP address
1.3.6.1.2.1.4.6	ipForwDatagrams	R	Number of received IP Frames that were passed on (routed)
1.3.6.1.2.1.4.7	ipUnknownProtos	R	Number of received IP Frames with an unknown protocol type
1.3.6.1.2.1.4.8	ipInDiscards	R	Number of received IP Frames that were rejected although no disturbances was present.
1.3.6.1.2.1.4.9	ipInDelivers	R	Number of received IP Frames that were passed on a higher protocol layer.
1.3.6.1.2.1.4.10	ipOutRequests	R	Number of sent IP Frames
1.3.6.1.2.1.4.11	ipOutDiscards	R	Number of rejected IP Frames that should have been sent
1.3.6.1.2.1.4.12	ipOutNoRoutes	R	Number of sent IP Frames that were rejected because of incorrect routing information.
1.3.6.1.2.1.4.13	ipReasmTimeout	R	Minimum time duration to a IP Frame is building up.
1.3.6.1.2.1.4.14	ipReasmReqds	R	Minimum number of the IP fragments for building up and pass on.
1.3.6.1.2.1.4.15	ipReasmOKs	R	Number successfully IP Frames re-assembled
1.3.6.1.2.1.4.16	ipReasmFails	R	Number not successfully IP Frames re-assembled
1.3.6.1.2.1.4.17	ipFragOKs	R	Number of IP Frames that were fragmented and passed on
1.3.6.1.2.1.4.18	ipFragFails	R	Number of IP Frames that had to be discarded because they need to be fragmented at this entity, but could not
1.3.6.1.2.1.4.19	ipFragCreates	R	Number of produced IP fragment Frames
1.3.6.1.2.1.4.20	ipAddrTable	-	Table of all local IP addresses of the coupler/controller
1.3.6.1.2.1.4.20.1	ipAddrEntry	-	Address information for an entry
1.3.6.1.2.1.4.20.1.1	ipAdEntAddr	R	The IP address those the address information concerns
1.3.6.1.2.1.4.20.1.2	ipAdEntIfIndex	R	Index of the interface
1.3.6.1.2.1.4.20.1.3	ipAdEntNetMask	R	The associated Subnet mask to the entry
1.3.6.1.2.1.4.20.1.4	ipAdEntBcastAddr	R	Value of the last significant bit in the IP broadcast address
1.3.6.1.2.1.4.20.1.5	ipAdEntReasmMaxSize	R	The size of the longest IP telegram that can be re-assembled again.
1.3.6.1.2.1.4.23	ipRoutingDis	R	Number of deleted Routing entries

	cards		

3.1.10.1.5 IpRoute Table

The IP Route Table contains information about the Routing table in the coupler/controller.

Identifier	Entry	Ac-cess	Description
1.3.6.1.2.1.4.21	ipRouteTable	-	IP Routing table
1.3.6.1.2.1.4.21.1	ipRouteEntry	-	A Routing entry for a special destination
1.3.6.1.2.1.4.21.1.1	ipRouteDest	R/W	This entry indicates the destination address of the Routing entry
1.3.6.1.2.1.4.21.1.2	ipRouteIfIndex	R/W	This entry indicates the index of the interface, which is the next route destination
1.3.6.1.2.1.4.21.1.3	ipRouteMetric1	R/W	The primary route to the target system
1.3.6.1.2.1.4.21.1.4	ipRouteMetric2	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.5	ipRouteMetric3	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.6	ipRouteMetric4	R/W	An alternative route to the target system
.3.6.1.2.1.4.21.1.7	ipRouteNextHop	R/W	The IP addresse of the next route section
1.3.6.1.2.1.4.21.1.8	ipRouteType	R/W	The kind of route
1.3.6.1.2.1.4.21.1.9	ipRouteProto	R	Routing mechanism via which the route is developed
1.3.6.1.2.1.4.21.1.10	ipRouteAge	R/W	Number of seconds since then the route was renewed or examined the latest time
1.3.6.1.2.1.4.21.1.11	ipRouteMask	R/W	This entry contents the subnet mask to this entry
1.3.6.1.2.1.4.21.1.12	ipRouteMetric5	R/W	An alternative route to the target system
1.3.6.1.2.1.4.21.1.13	ipRouteInfo	R/W	A reference to a special MIB

3.1.10.1.6 IpNetToMediaTable

Identifier	Entry	Ac-cess	Description
1.3.6.1.2.1.4.22	ipNetToMedia Table	-	Relocation dictionary for the allocation from IP addresses to hardware addresses
1.3.6.1.2.1.4.22.1	ipNetToMedia Entry	-	Entry of the table described above
1.3.6.1.2.1.4.22.1.1	ipNetToMediaIf Index	R/W	Index for the interface
1.3.6.1.2.1.4.22.1.2	ipNetToMedia PhysAddress	R/W	The hardware address of the interface
1.3.6.1.2.1.4.22.1.3	ipNetToMedia NetAddress	R/W	The IP address of the interface
1.3.6.1.2.1.4.22.1.4	ipNetToMedia Type	R/W	Kind of mapping

3.1.10.1.7 ICMP Group

Identifier	Entry	Ac-cess	Description
1.3.6.1.2.1.5.1	icmpInMsgs	R	Number of received ICMP messages
1.3.6.1.2.1.5.2	icmpInErrors	R	Number of received ICMP messages that

			contain the ICMP specific errors
1.3.6.1.2.1.5.3	icmpInDestUnreachs	R	Number of received ICMP destination unreachable messages
1.3.6.1.2.1.5.4	icmpInTimeExcds	R	Number of received ICMP time exceeded messages
1.3.6.1.2.1.5.5	icmpInParmProbs	R	Number of received ICMP parameter problems messages
1.3.6.1.2.1.5.6	icmpInSrcQuenchs	R	Number of received ICMP source quench messages
1.3.6.1.2.1.5.7	icmpInRedirects	R	Number of received ICMP redirect messages
1.3.6.1.2.1.5.8	icmpInEchos	R	Number of received ICMP echo request messages (Ping)
1.3.6.1.2.1.5.9	icmpInEchoReps	R	Number of received ICMP echo reply messages (Ping)
1.3.6.1.2.1.5.10	icmpInTimestamps	R	Number of received ICMP timestamp request messages
1.3.6.1.2.1.5.11	icmpInTimestampReps	R	Number of received ICMP timestamp reply messages
1.3.6.1.2.1.5.12	icmpInAddrMasks	R	Number of received ICMP address mask request messages
1.3.6.1.2.1.5.13	icmpInAddrMaskReps	R	Number of received ICMP address mask reply messages
1.3.6.1.2.1.5.14	icmpOutMsgs	R	Number of sent ICMP messages
1.3.6.1.2.1.5.15	icmpOutErrors	R	Number of sent ICMP messages that could not to be sent because of problems
1.3.6.1.2.1.5.16	icmpOutDestUnreachs	R	Number of sent ICMP destination unreachable messages
1.3.6.1.2.1.5.17	icmpOutTimeExcds	R	Number of sent ICMP time exceeded messages
1.3.6.1.2.1.5.18	icmpOutParmProbs	R	Number of sent ICMP parameter problem messages
1.3.6.1.2.1.5.19	icmpOutSrcQuenchs	R	Number of sent ICMP source quench messages
1.3.6.1.2.1.5.20	icmpOutRedirects	R	Number of sent ICMP redirection messages
1.3.6.1.2.1.5.21	icmpOutEchos	R	Number of sent ICMP echo request messages
1.3.6.1.2.1.5.22	icmpOutEchoReps	R	Number of sent ICMP echo reply messages
1.3.6.1.2.1.5.23	icmpOutTimestamps	R	Number of sent ICMP timestamp request messages
1.3.6.1.2.1.5.24	icmpOutTimestamp Reps	R	Number of sent ICMP timestamp reply messages
1.3.6.1.2.1.5.25	icmpOutAddrMasks	R	Number of sent ICMP address mask request messages
1.3.6.1.2.1.5.26	icmpOutAddrMask Reps	R	Number of sent ICMP address mask reply messages

3.1.10.1.8 TCP Group

Identifier	Entry	Ac-cess	Description
1.3.6.1.2.1.6.1	tcpRtoAlgorithm	R	Retransmission time (1 = another, 2 = constant, 3 = MIL standard 1778, 4 = Jacobson)
1.3.6.1.2.1.6.2	tcpRtoMin	R	Minimum value for the retransmission timer
1.3.6.1.2.1.6.3	tcpRtoMax	R	Maximum value for the retransmission timer
1.3.6.1.2.1.6.4	tcpMaxConn	R	Number of maximum TCP connections that can exist at the same time
1.3.6.1.2.1.6.5	tcpActiveOpens	R	Number of existing active TCP connections
1.3.6.1.2.1.6.6	tcpPassiveOpens	R	Number of existing passive TCP connections
1.3.6.1.2.1.6.7	tcpAttemptFails	R	Number of missed connection establishment attempts
1.3.6.1.2.1.6.8	tcpEstabResets	R	Number of connection resets
1.3.6.1.2.1.6.9	tcpCurrEstab	R	Number of TCP connections in the established or close-wait status
1.3.6.1.2.1.6.10	tcpInSegs	R	Number of received TCP frames including the error frames
1.3.6.1.2.1.6.11	tcpOutSegs	R	Number of correctly sent TCP frames with data
1.3.6.1.2.1.6.12	tcpRetransSegs	R	Number of sent TCP frames that were repeated because of errors
1.3.6.1.2.1.6.13	tcpConnTable	-	For each existing connection a table entry is produced
1.3.6.1.2.1.6.13.1	tcpConnEntry	-	Table entry for connection
1.3.6.1.2.1.6.13.1.1	tcpConnState	R	The entry indicates the status of the TCP connection
1.3.6.1.2.1.6.13.1.2	tcpConnLocal Address	R	The entry contains the IP address for the connection. For a server, this entry ist constant 0.0.0.0
1.3.6.1.2.1.6.13.1.3	tcpConnLocalPort	R	The entry indicates the port number of the TCP connection.
1.3.6.1.2.1.6.13.1.4	tcpConnRem Address	R	The entry contains the remote IP address of the TCP connection.
1.3.6.1.2.1.6.13.1.5	tcpConnRemPort	R	The entry contains the remote port of the TCP connection.
1.3.6.1.2.1.6.14	tcpInErrs	R	Number of received incorrect TCP frame
1.3.6.1.2.1.6.15	tcpOutRsts	R	Number of sent TCP frames with set RST flag

3.1.10.1.9 UDP Group

Identifier	Entry	Ac-cess	Description
1.3.6.1.2.1.7.1	udpInDatagrams	R	Number of received UDP frames that could be passed on the appropriate applications
1.3.6.1.2.1.7.2	udpNoPorts	R	Number of received UDP frames that could not be passed on the appropriate applications (port unreachable)

1.3.6.1.2.1.7.3	udpInErrors	R	Number of received UDP frames that could not be passed on the appropriate applications for other reasons.
1.3.6.1.2.1.7.4	udpOutDatagrams	R	Number of sent UDP frames
1.3.6.1.2.1.7.5	udpTable	-	For each application, which received UDP frames, a table entry is produced
1.3.6.1.2.1.7.5.1	udpEntry	-	Table entry for an application that received an UDP Frame
1.3.6.1.2.1.7.5.1.1	udpLocalAddress	R	IP address of the local UDP server
1.3.6.1.2.1.7.5.1.2	udpLocalPort	R	Port number of the local UDP server

3.1.10.1.10 SNMP Group

Identifier	Entry	Ac-cess	Description
1.3.6.1.2.1.11.1	snmpInPkts	R	Number of received SNMP frames
1.3.6.1.2.1.11.2	snmpOutPkts	R	Number of sent SNMP frames
1.3.6.1.2.1.11.3	snmpInBadVersions	R	Number of received SNMP frames with an invalid version number
1.3.6.1.2.1.11.4	snmpInBadComm unityNames	R	Number of received SNMP frames with an invalid community
1.3.6.1.2.1.11.5	snmpInBadComm unityUses	R	Number of received SNMP frames of which the community did not have a sufficient authorization for the actions that it tried to accomplished
1.3.6.1.2.1.11.6	snmpInASNParseErrs	R	Number of received SNMP frames, which had a wrong structure
1.3.6.1.2.1.11.8	snmpInTooBigs	R	Number of received SNMP frames that acknowledged the result too Big
1.3.6.1.2.1.11.9	snmpInNoSuchNames	R	Number of received SNMP frames that acknowledged the result noSuchName
1.3.6.1.2.1.11.10	snmpInBadValues	R	Number of received SNMP frames that acknowledged the result bad value
1.3.6.1.2.1.11.11	snmpInReadOnlys	R	Number of received SNMP frames that acknowledged the result readOnly
1.3.6.1.2.1.11.12	snmpInGenErrs	R	Number of received SNMP frames that acknowledged the result genError
1.3.6.1.2.1.11.13	snmpInTotalReqVars	R	Number of received SNMP frames with valid GET- or GET-NEXT requests
1.3.6.1.2.1.11.14	snmpInTotalSetVars	R	Number of received SNMP frames with valid SET requests
1.3.6.1.2.1.11.15	snmpInGetRequests	R	Number of received and implemented GET requests
1.3.6.1.2.1.11.16	snmpInGetNexts	R	Number of received and implemented GET-NEXT requests
1.3.6.1.2.1.11.17	snmpInSetRequests	R	Number of received and implemented SET requests
1.3.6.1.2.1.11.18	snmpInGetResponses	R	Number of received GET responses
1.3.6.1.2.1.11.19	snmpInTraps	R	Number of received Traps
1.3.6.1.2.1.11.20	snmpOutTooBigs	R	Number of sent SNMP frames that contained the result too Big

1.3.6.1.2.1.11.21	snmpOutNoSuch Names	R	Number of sent SNMP frames that contained the result noSuchName
1.3.6.1.2.1.11.22	snmpOutBadValues	R	Number of sent SNMP frames that contained the result bad value
1.3.6.1.2.1.11.24	SnmpOutGenErrs	R	Number of sent SNMP frames that contained the result genErrs
1.3.6.1.2.1.11.25	snmpOutGetRequests	R	Number of sent GET requests
1.3.6.1.2.1.11.26	SnmpOutGetNexts	R	Number of sent GET NEXT requests
1.3.6.1.2.1.11.27	snmpOutSetRequests	R	Number of sent SET requests
1.3.6.1.2.1.11.28	snmpOutGetResponses	R	Number of sent GET responses
1.3.6.1.2.1.11.29	snmpOutTraps	R	Number of sent Traps
1.3.6.1.2.1.11.30	snmpEnableAuthen Traps	R/W	Authentification failure Traps (1 = on, 2 = off)

3.1.10.1.11 EGP-Group

This group contains information of the EGP (Exterior Gateway Protocol) protocol layer. This protocol is used mainly when routing the Internet provider for the Internet binding. This group is not supported by some couplers/controllers.

3.1.10.2 Traps

With certain events the SNMP agent can dispatch independently (without inquiry by the manager) one of the following messages:

coldStart	Restart of the component
authenticationFailure	Unauthorized (missed) MIB access
enterpriseSpecific	tbd

3.1.11 LED Display

The controller has several LED's for a visual display of the controller and nodes operating status.

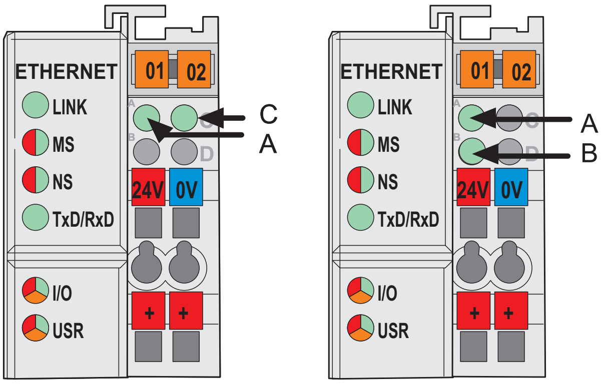


Fig. 3.1-15: Display elements 750-841 g084102x

The LEDs can be divided into three groups.

The first group of LEDs display the status of the Ethernet fieldbus. It contains both solid and two-color LEDs. They are labelled as: 'LINK' (green), 'MS' (red/green), 'NS' (red/green), and 'TxD/RxD' (green).

The second group of LEDs are three-color LEDs (red/green/orange). One of the LED is labelled 'I/O', and displays the status of the internal bus. The other is labelled 'USR', and is programmable with WAGO-I/O-*PRO* CAA.

The third group uses solid colored LEDs. They are located on the right-hand side of the controller power supply. These display the status of the supply voltage.

3.1.11.1 Fieldbus status

The health of the ETHERNET Fieldbus is signalled through the top LED group (‘LINK‘, ‘MS‘, ‘NS‘ and ‘TxD/RxD‘). The two-colored LEDs ‘MS’ (module status) and ‘NS’ (network status) are solely used by the Ethernet/IP protocol. These two LEDs conform to the Ethernet/IP specifications.

LED	Meaning	Trouble shooting
LINK		
green	Link to a physical network exists	
OFF	No link to a physical network	Check the fieldbus connection.
MS		
red / green flashing	Self test	
red	The system indicates a not remediable error	Restart the fieldbus coupler by turning the power supply off and on again. If the error still exists, please contact the I/O Support.
green flashing	The system is not yet configures	
green	Normal operation	
OFF	No system supply voltage	Check the supply voltage (24V and 0V)
NS		
red / green flashing	Self test	
red	The system indicates a double IP-address in the network	Use an IP address that is not used yet.
red flashing	At least one connection (MODBUS/TCP or Ethernet/IP) announced a Timeout, where the CONTROLLER functions as target.	Restart the fieldbus coupler by turning the power supply off and on again and develop a new connection.
green flashing	No connection (MODBUS/TCP or Ethernet/IP).	
green	At least one connection (MODBUS/TCP or Ethernet/IP) is developed (also connection to the Message rout applies)	
OFF	No IP address is assigned to the system.	Assign to the system an IP address by BootP, DHCP or the Ethernet Settings tool.
TxD/RxD		
green	Data exchange via ETHERNET taking place	
OFF	No data exchange via ETHERNET	

3.1.11.2 Node Status – Blink code from the 'I/O' LED

The ‘I/O‘-LED displays the communication status of the internal bus. Additionally, this LED is used to display fault codes (blink codes) in the event of a system error.

LED	Meaning	Trouble shooting
I/O		
Green	Fieldbus controller operating perfectly, Data cycle on the internal bus	
Off	No data cycle on the internal bus	
Red	a) During startup of fieldbus controller: Internal bus being initialized, Startup displayed by LED flashing fast for approx. 1-2 seconds	
Red	b) After startup of fieldbus controller: Errors, which occur, are indicated by three consecutive flashing sequences. There is a short pause between each sequential flash.	Evaluate the fault message (fault code and fault argument).

The controller starts up after switching on the supply voltage. The "I/O" LED blinks. The "I/O" LED has a steady light following a fault free run-up.
In the case of a fault the "I/O" LED continues blinking. The fault is cyclically displayed by the blink code.

Detailed fault messages are displayed with the aid of a blink code. A fault is cyclically displayed with up to 3 blink sequences.

- The first blink sequence (approx. 10 Hz) starts the fault display.
- The second blink sequence (approx. 1 Hz) following a pause. The number of blink pulses indicates the **fault code**.
- The third blink sequence (approx. 1 Hz) follows after a further pause. The number of blink pulses indicates the **fault argument**.

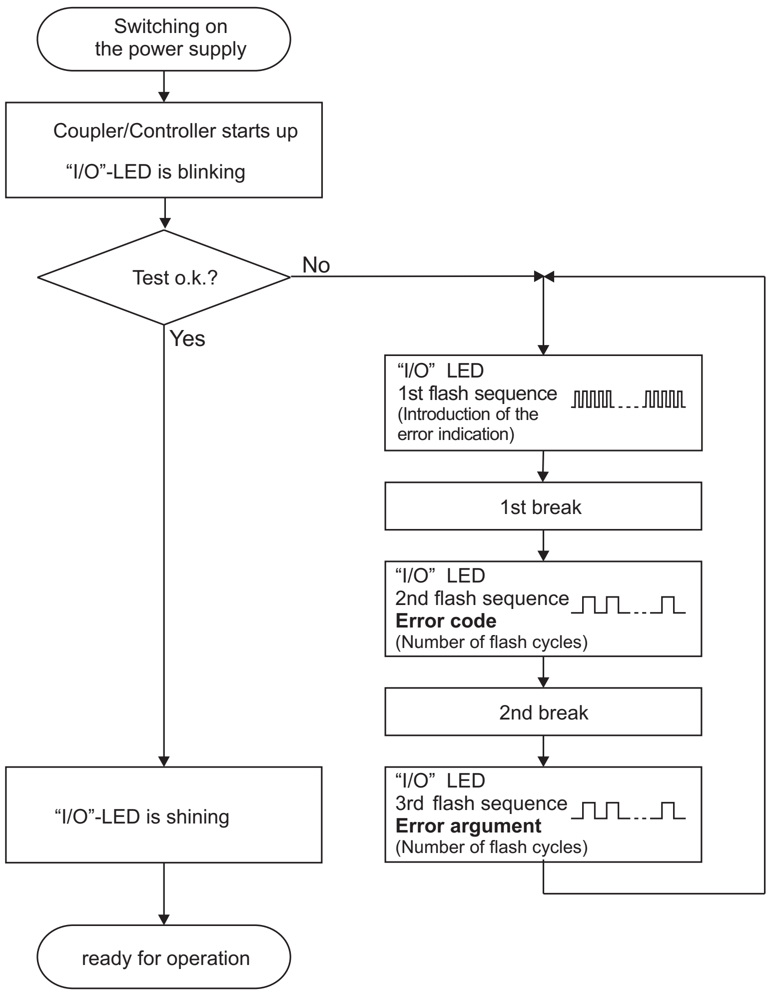


Fig. 3.1-16: Signalling of the LED for indication of the node status
g012111e

After clearing a fault, restart the controller by cycling the power.

Fault message of the 'I/O' LED

1st blink sequence: Start of the Fault message
2nd blink sequence: Fault code
3rd blink sequence: Fault argument

Fault code 1: " Hardware and configuration fault "		
Fault argument	**Fault description**	**Trouble shooting**
1	Overflow of the internal buffer memory for the inline code	Turn off the power supply of the node, reduce number of I/O modules and turn the power supply on again. If the error still exists, exchange the bus coupler.

2	I/O module(s) with unsupported data type	Detect faulty I/O module as follows: turn off the power supply. Place the end module in the middle of the fieldbus node. Turn the power supply on again. – If the LED is still blinking, turn off the power supply and place the end module in the middle of the first half of the node (towards the coupler). – If the LED doesn’t blink, turn off the power supply and place the end module in the middle of the second half of the node (away from the coupler). Turn the power supply on again. Repeat this procedure until the faulty I/O module is detected. Replace the faulty I/O module. Ask about a firmware update for the fieldbus coupler.
3	Invalid fieldbus coupler/controller parameter checksum	Turn off the power supply of the node, reduce number of I/O modules and turn the power supply on again
4	Error occurred when writing to serial EEPROM	Turn off the power supply of the node, exchange fieldbus coupler and turn the power supply on again.
5	Error occurred with read access to serial EEPROM	Turn off the power supply of the node, exchange fieldbus coupler and turn the power supply on again.
6	Changed I/O module configuration determined after AUTORESET	Restart the fieldbus coupler by turning the power supply off and on again.
7	Firmware does not run on existing hardware	Turn off the power supply of the node, exchange fieldbus coupler and turn the power supply on again.
8	Time limit exceeded for accessing the serial EEPROM	Turn off the power supply of the node, exchange fieldbus coupler and turn the power supply on again.
9	Bus Coupler initialisation fault	Turn off the power supply of the node, exchange fieldbus coupler and turn the power supply on again.
10	RTC-Powerfail	Adjust the clock and keep upright the supply voltage of the bus coupler for at least 15 minutes for loading of the Goldcaps.

11	Fault when reading out the time from the RTC	Adjust the clock and keep upright the supply voltage of the bus coupler for at least 15 minutes for loading of the Goldcaps.
12	Fault when writing the time in the RTC	Adjust the clock and keep upright the supply voltage of the bus coupler for at least 15 minutes for loading of the Goldcaps.
13	Error Clock-Interrupt	Adjust the clock and keep upright the supply voltage of the bus coupler for at least 15 minutes for loading of the Goldcaps.
14	Maximum number of Gateway or Mailbox I/O modules exceeded	Turn off the power supply of the node, reduce number of Gateway or Mailbox I/O modules and turn the power supply on again.
Fault code 2: -not used-		
Fault argument	**Fault description**	**Trouble shooting**
-	not used	-

Fault code 3: "Internal bus protocol fault"		
Fault argument	**Fault description**	**Trouble shooting**
-	Internal bus communication malfunction; faulty device can’t be detected	If the fieldbus node comprises internal system supply modules (750-613), make sure first that the power supply of these modules is functioning. This is indicated by the status LEDs. If all I/O modules are connected correctly or if the fieldbus node doesn’t comprise 750-613 modules you can detect the faulty I/O module as follows: turn off the power supply of the node. Place the end module in the middle of the fieldbus node. Turn the power supply on again. – If the LED is still blinking, turn off the power supply and place the end module in the middle of the first half of the node (towards the coupler). – If the LED doesn’t blink, turn off the power supply and place the end module in the middle of the second half of the node (away from the coupler). Turn the power supply on again. Repeat this procedure until the faulty I/O module is detected. Replace the faulty I/O module. If there is only one I/O module left but the LED is still blinking, then this I/O module or the coupler is defective. Replace defective component.

Fault code 4: "Internal bus physical fault"		
Fault argument	**Fault description**	**Trouble shooting**
-	Error in internal bus data communication or interruption of the internal bus at the coupler	Turn off the power supply of the node. Place an I/O module with process data behind the coupler and note the error argument after the power supply is turned on. If no error argument is given by the I/O LED, replace the coupler. Otherwise detect faulty I/O module as follows: turn off the power supply. Place the end module in the middle of the fieldbus node. Turn the power supply on again. – If the LED is still blinking, turn off the power supply and place the end module in the middle of the first half of the node (towards the coupler). – If the LED doesn't blink, turn off the power supply and place the end module in the middle of the second half of the node (away from the coupler). Turn the power supply on again. Repeat this procedure until the faulty I/O module is detected. Replace the faulty I/O module. If there is only one I/O module left but the LED is still blinking, then this I/O module or the coupler is defective. Replace defective component.
n*	Interruption of the internal bus after the n^{th} process data module.	Turn off the power supply of the node, exchange the $(n+1)^{th}$ process data module and turn the power supply on again.
Fault code 5: "Internal bus initialization fault"		
Fault argument	**Fault description**	**Trouble shooting**
n*	Error in register communication during internal bus initialization	Turn off the power supply of the node and replace n^{th} process data module and turn the power supply on again.

Fault code 6: "Fieldbus specific errors"		
Fault argument	**Fault description**	**Trouble shooting**
1	Invalid MACID	Turn off the power supply of the node, exchange fieldbus coupler and turn the power supply on again.
2	Ethernet Hardware initialization error	Restart the fieldbus coupler by turning the power supply off and on again. If the error still exists, exchange the bus coupler.
3	TCP/IP initialization error	Restart the fieldbus coupler by turning the power supply off and on again. If the error still exists, exchange the bus coupler.
4	Network configuration error (no IP Address)	Check the settings of BootP server.
5	Application protocol initialization error	Restart the fieldbus coupler by turning the power supply off and on again.
6	Process image is too large	Reduce number of I/O modules
7	Double IP address in network	Use another IP adresse, which is not yet present in network.
8	Error when building the process image	Reduce number of I/O modules
Fault code 9: -not used-		
Fault argument	**Fault description**	**Trouble shooting**
-	not used	-
Fault code 10: "PLC program fault "		
Fault argument	**Fault description**	**Trouble shooting**
1	Error when implementing the PFC run time system	Restart the fieldbus coupler by turning the power supply off and on again. If the error still exists, please contact the I/O Support.
2	Error when generating the PFC inline code	Restart the fieldbus coupler by turning the power supply off and on again. If the error still exists, please contact the I/O Support.
3	An IEC task exceeded the maximum running time or the sampling interval of the IEC task could not be kept (Watchdog)	Check the task configuration concerning the adjusted sampling intervals and watchdog times.
4	PFC Web-Visualization initialization error	Restart the fieldbus coupler by turning the power supply off and on again. If the error still exists, please accomplish a reset (origin) in WAGO-I/O-PRO, compile the project again and transfer it to the coupler.

Fault code 11: "Gateway-/Mailbox I/O module fault"		
Fault argument	**Fault description**	**Trouble shooting**
1	Maximum number of Gateway modules exceeded	Turn off the power supply of the node, reduce number of Gateway modules and turn the power supply on again.
2	Maximum size of Mailbox exceeded	Reduce the Mailbox size.
3	Maximum size of process image exceeded due to the put Gateway modules	Reduce the data width of the Gateway modules.

* The number of blink pulses (n) indicates the position of the I/O module. I/O modules without data are not counted (e.g. supply module without diagnosis)

Example: The 13th I/O module has been removed.	
1.	The I/O-LED starts the fault display with the first blink sequence (approx. 10 Hz).
2.	The second blink phase (approx. 1 Hz) follows the first pause. The I/O-LED blinks four times and thus signals the fault code 4 (internal bus data fault).
3.	The third blink sequence follows the second pause. The I/O-LED blinks twelve times. The fault argument 12 means that the internal bus is interrupted after the 12^{th} I/O module.

3.1.11.3 ‘USR‘-LED

The state of the ‘USR‘ LED is programmable with WAGO-I/O-*PRO* CAA. Functions in the program library ”Visual.lib“ can be used to control the LED state. One of the many possible uses of this LED is to indicate the RUN/STOP state of your controller.

3.1.11.4 Supply voltage status

The two green LED’s in the controller supply section, display the status of the supply voltage. The left LED (A) indicates the status of the 24 V supply for the coupler. The other LED (‘B‘ or ‘C‘) displays the status of the field side supply (i.e., the power jumper contacts).

LED	Meaning	Trouble shooting
A		
Green	Operating voltage for the system exists.	
OFF	No operating voltage for the system.	Check the supply voltage (24V and 0V).
B or C		
Green	Operating voltage for the power jumper contacts exists.	
OFF	No operating voltage for the the power jumper contacts.	Check the supply voltage (24V and 0V).

3.1.12 Fault behavior

3.1.12.1 Fieldbus failure

When a Modbus TCP fieldbus failure occurs (e.g., the Ethernet cable is removed or broke), the outputs that are controlled by the fieldbus port remain in their current state by default. If this behavior is undesirable, a fieldbus watchdog timer can be programmed to monitor Fieldbus communications. The watchdog monitors the data transfer between the master controls and the PFC. In the event of a watchdog timeout (i.e., there is a Fieldbus failure), the PFC can be programmed to control the state of the outputs, based on your application needs. In the case of fault free communications, the watchdog timer will not timeout, since after each successful data transfer the watchdog timer is reset.

Monitoring the watchdog time in the PFC is done by using the function block 'FBUS_ERROR_INFORMATION' in the control program. This function is part of the “mod_com.lib” library.

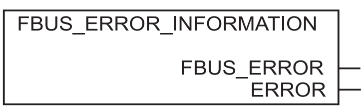


Fig. 3.1-17: Function block for determining a fieldbus failure g012926x

'FBUS_ERROR' (BOOL) = FALSE = no fault

= TRUE = fieldbus failure

'ERROR' (WORD) = 0 = no fault
= 1 = fieldbus failure

More information
For detailed information to the Watchdog register see the Chapters "MODBUS Functions"; "Watchdog (Fieldbus failure)" and "Watchdog Register".

3.1.12.2 Internal bus fault

When an internal bus fault occurs (e.g., an I/O module is removed), all output modules turn off. Additionally, the "I/O" LED blinks red and generates a fault message. This message is decoded from the blink code, in the form of a fault code and fault argument.

Once the internal bus fault is fixed, the controller is restarted by cycling its power. At this point the transfer of the process data resumes and the outputs are updated.

3.1.13 Technical Data

System data	
No. of nodes	limited by ETHERNET specification
Transmission medium	Twisted Pair S-UTP 100 Ω cat. 5
Buscoupler connection	RJ45
Max. length of fieldbus segment	100 m between hub station and 750-841; max. length of network limited by ETHERNET specification
Baud rate	10/100 Mbit/s
Protocols	MODBUS/TCP (UDP), ETHERNET/IP, HTTP, BootP, DHCP, DNS, SNTP, FTP, SNMP
Programming	WAGO-I-PRO CAA
IEC 61131-3-3	IL, LD, FBD, ST, SFC
Technical Data	
No. of I/O modules with bus extension	64 250
Fieldbus Input process image max. Output process image max. Input variables max Output variables max	 2 kByte 2 kByte 512 Byte 512 Byte
Configuration possibility	via PC
Program memory	512 kByte
Data memory	256 kByte
Non-valatile memory	24 kByte (16 k retain, 8 k flags)
Max. no. of socket connections	3 HTTP, 15 MODBUS/TCP, 10 FTP, 2 SNMP, 5 for IEC 61131-3 programs, 2 for WAGO-I/O-*PRO* CAA, 128 for EtherNet/IP
Powerfail-RTC-Buffer	min. 6 days
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 I/O modules	1700 mA at 5 V
Isolation	500 V system/supply
Voltage via power jumper contacts	DC 24 V (-25 % / + 30 %)
Current via power jumper contacts$_{max}$	DC 10 A
Dimensions (mm) W x H x L	51 x 65* x 100 (*from upper edge of DIN 35 rail)
Weight	ca. 184 g
Accessories	
WAGO-I/O-*PRO* 32 or WAGO-I/O-*PRO* CAA	759-332 759-333
Miniature WSB quick marking system	
Standards and Regulations (cf. Chapter 2.2)	
EMC CE-Immunity to interference	acc. to EN 61000-6-2 (99)
EMC CE-Emission of interference	acc. to EN 50081-2 (94)
EMC marine applications-Immunity to interference	acc. to Germanischer Lloyd (2001)
EMC marine applications-Emission of	acc. to Germanischer Lloyd (2001)

interference		
Approvals (cf. Chapter 2.2)		
	$_{C}$UL$_{US}$ (UL508)	
	ABS (American Bureau of Shipping) [1]	
	BV (Bureau Veritas) [1]	
	GL (Germanischer Lloyd) [1]	Cat. A, B, C, D
	KR (Korean Register of Shipping) [1]	
	LR (Lloyd's Register) [1]	Env. 1, 2, 3, 4
NKK	NKK (Nippon Kaiji Kyokai) [1]	
CE	Conformity Marking	
[1] Consider chapter: „Supplementary power supply regulations”!		

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, 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.

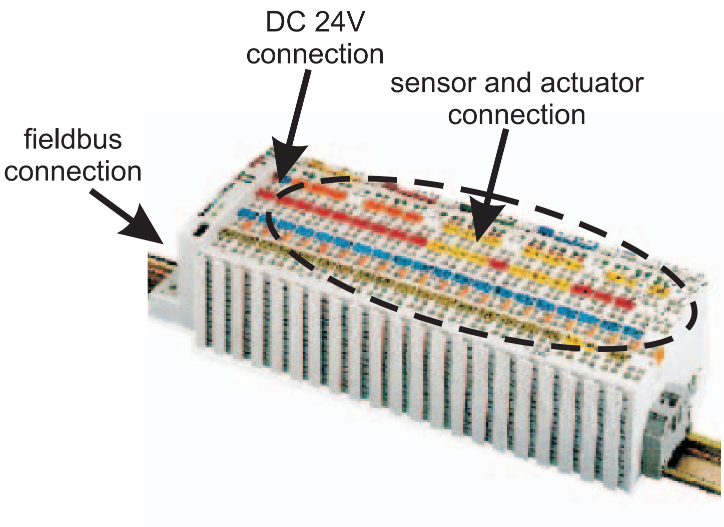


Fig. 4-1. Connection Example and Principle of a Fieldbus Node for a Network Architecture
1Netwerkknotene

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).

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).
10Base-F	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: 10Base-FL for fiber-optic link, 10Base-FB for fiber-optic backbone and 10Base-FP for fiber-optic passive).
10Base-T	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.

Tab. 4-1: ETHERNET Transmission Standards

Beyond that there are still further transmission standards, for example: 100Base-T4 (Fast ETHERNET over twisted conductors), 100Base-FX (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.

10Base-T, 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.

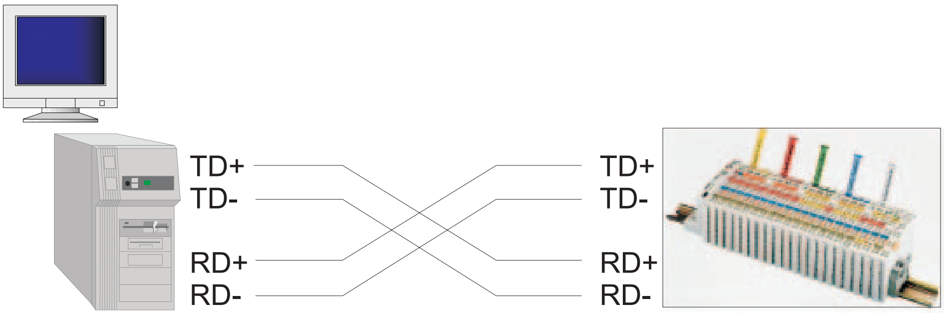


Fig. 4-2: Direct Connection of a Node with Crossover Cable g012906d

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.

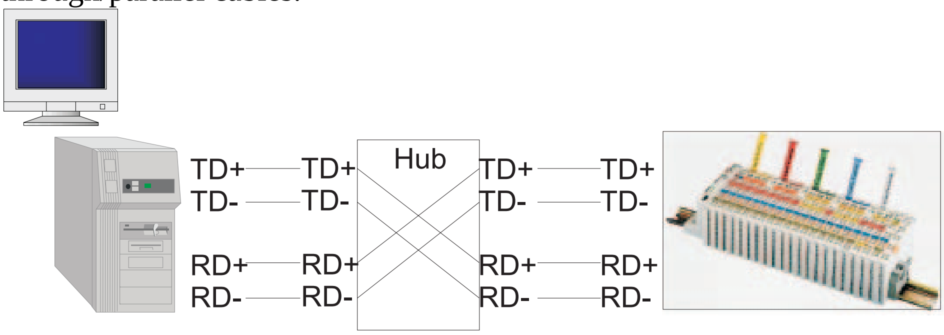


Fig. 4-3: Connection of a Node by means of a Hub with Parallel cables g012908d

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 10Base-T, or 100BaseTX several stations (nodes) are connected using a star topology according to the 10Base-T 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.

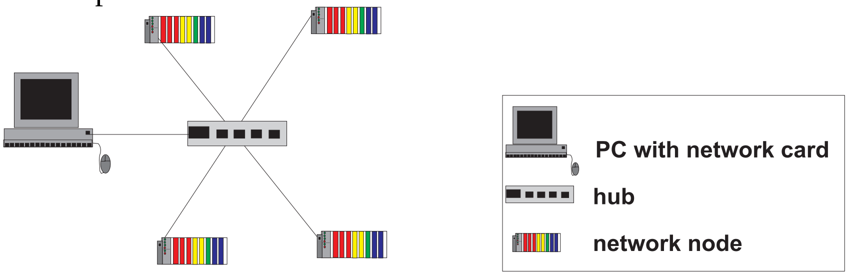


Fig. 4-4: Star Topology G012903e

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.

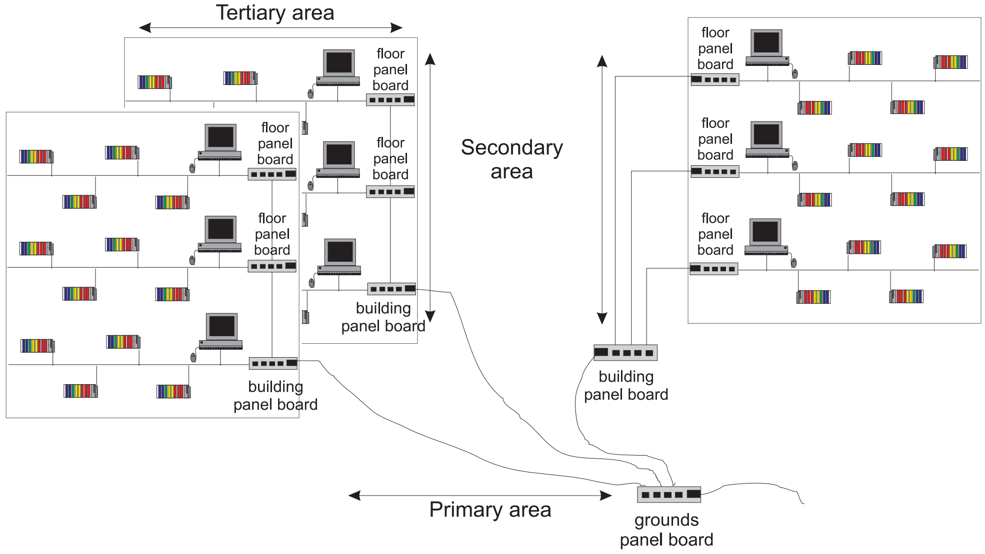


Fig. 4-5: Tree Topology G012904e

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.

Note
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.

Note
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.

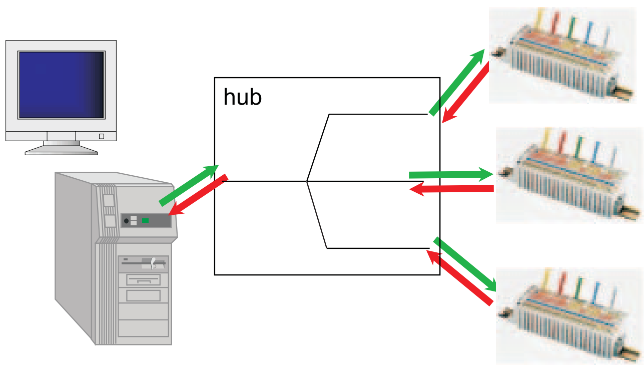


Fig. 4-6: Principle of Shared ETHERNET G012910e

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.

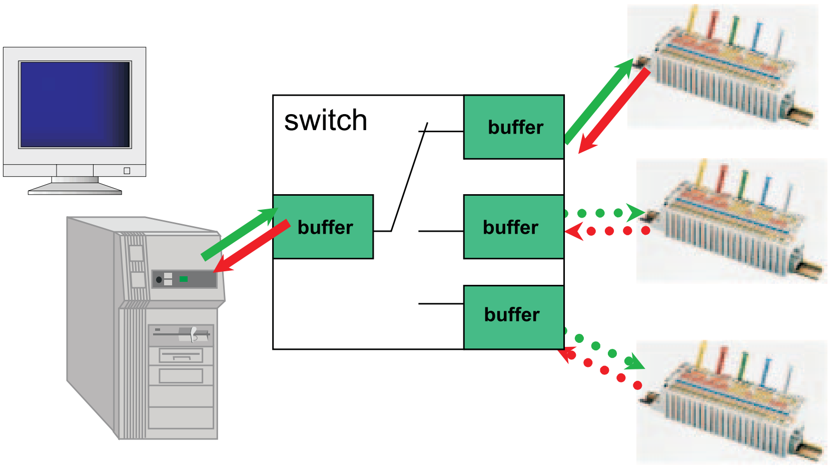


Fig. 4-7: Principle of Switched ETHERNET G012909e

4.1.3 Network Communication

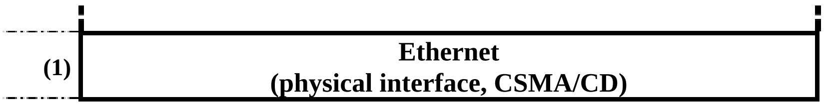
Fieldbus communication between master application and (programmable) fieldbus coupler or controller usually takes place using an implemented fieldbus specific application protocol, e. g. MODBUS TCP (UDP), EtherNet/IP, BACnet, KNXNET/IP, PROFINET, Powerlink, Sercos III or others.

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

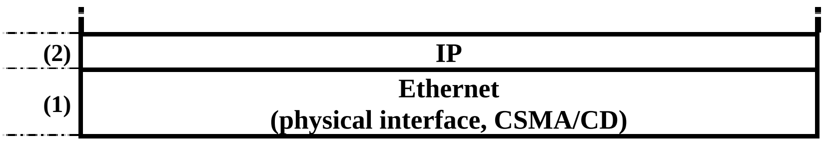
(1) Ethernet:

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.

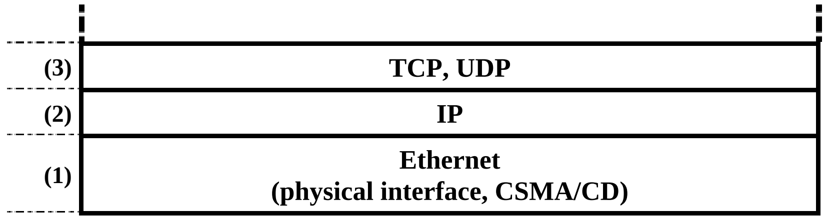


(3) TCP, UDP:

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.

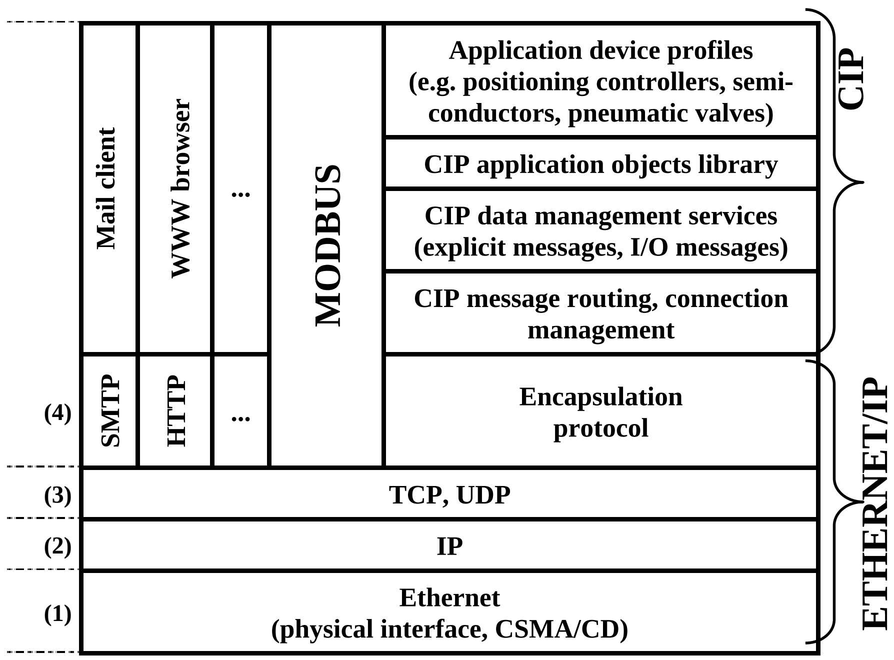
b) UDP: (User Datagram Protocol)
The UDP layer is also a transport protocol like TCP, and is arranged above the IP layer. In contrast to the TCP protocol, UDP is not connection oriented. That means there are no monitoring mechanisms for data exchange between sender and receiver.

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.

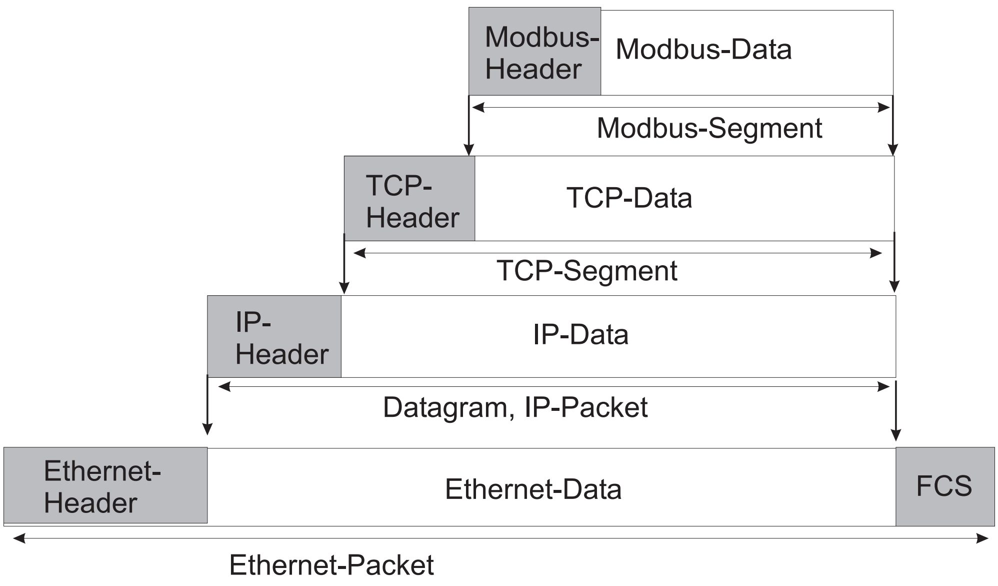


Fig. 4-8: Communication Protocols G012907e

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.

Note
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).

4.1.3.3 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.

4.1.3.3.1 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

<table>
<tr><td colspan="2">01100101</td><td>00010000</td><td>11101000</td><td>00010110</td></tr>
<tr><td>0</td><td>Net-ID</td><td colspan="3">Host-ID</td></tr>
</table>

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.

4.1.3.3.1.1 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.

4.1.3.3.1.2 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 realised with the help of the *Internet Group Message Protocol* IGMP; neighbouring 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.

4.1.3.3.2 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.

4.1.3.3.3 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.

4.1.3.3.4 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.

4.1.3.4 Administration and Diagnosis Protocols

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

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

More information

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

4.1.3.4.1 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.

More 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.

4.1.3.4.2 HTTP (HyperText Transfer Protocol)

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

Today, HTTP forms the basis of the Internet and is also based on requests and responses in the same way as the BootP protocol.

The HTTP server implemented in the (programmable) fieldbus coupler or controller is used for viewing the HTML pages saved in the coupler/controller.
The HTML pages provide information about the coupler/controller (state, configuration), the network and the process image.
On some HTML pages, (programmable) fieldbus coupler or controller settings can also be defined and altered via the web-based management system (e.g. whether IP configuration of the coupler/controller is to be performed via the DHCP protocol, the BootP protocol or from the data stored in the EEPROM).
The HTTP server uses port **number 80**.

4.1.3.4.3 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 initialisation. 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.

4.1.3.4.4 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.

4.1.3.4.5 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.

4.1.3.4.6 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.

Note
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
NLST	Gives the directory list
RMD	Deletes directory
PWD	Gives the actually path
MKD	Puts on a dirctory

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

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

4.1.3.4.7 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 outbasket.

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.

4.1.3.5 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
- KNXnet/IP
- PROFINET
- Powerlink
- Sercos III

More 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 MODBUS Functions

4.2.1 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.
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. 4-12: MODBUS/TCP Header

More 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.

More information
More information on the *OPEN MODBUS / TCP SPECIFICATION* you can find in the Internet: www.modbus.org.

Therefore the MODBUS protocol based essentially on the following basic data types:

Datatype	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 „FunctionCodes“ 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.

4.2.2 Use of the MODBUS Functions

The example below uses a graphical view of a fieldbus node to show which MODBUS functions can be used to access data of the process image.

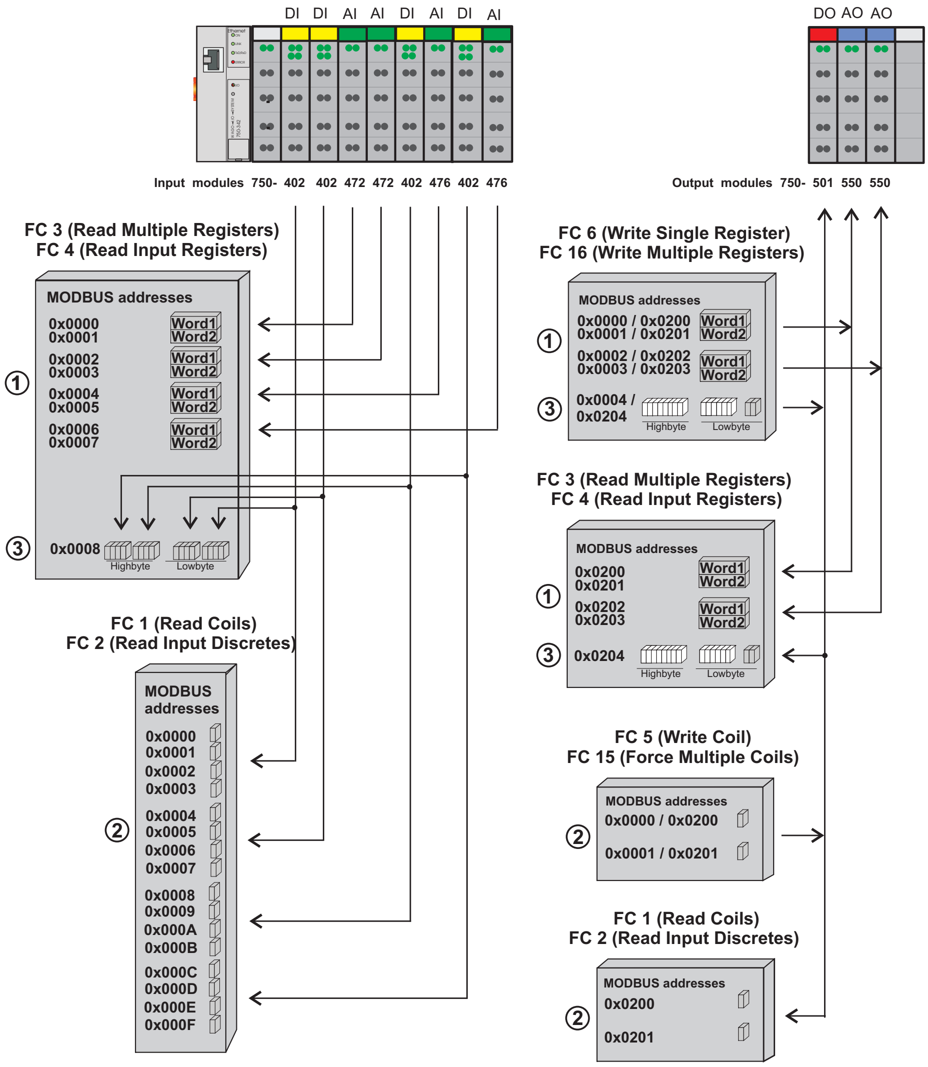


Fig. 4-13: Use of the MODBUS Functions G012918e

Attention
It is recommended that analog data be accessed with register functions (1) and digital data with coil functions (2).

4.2.3 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} - $62FC_{hex}$].

4.2.3.1 Function Code FC1 (Read Coils)

This function reads the status of the input and output bits (coils) in a slave device.

Request

The request specifies the reference number (starting address) and the bit count to read.
Example: Read output bits 0 to 7.

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

4.2.3.2 Function Code FC2 (Read Input Discretes)

This function reads the input bits from a slave device.

Request

The request specifies the reference number (starting address) and the bit count to be read.
Example: Read input bits 0 to 7:

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

4.2.3.3 Function Code FC3 (Read multiple registers)

This function reads the contents of holding registers from a slave device in word format.

Request

The request specifies the reference number (start register) and the word count (register quantity) of the registers to be read. The reference number of the request is zero based, therefore, the first register starts at address 0.
Example: Read registers 0 and 1:

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

4.2.3.4 Function code FC4 (Read input registers)

This function reads contents of input registers from the slave device in word format.

Request

The request specifies a reference number (start register) and the word count (register quantity) of the registers to be read. The reference number of the request is zero based, therefore, the first register starts at address 0.

Example: Read registers 0 and 1:

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

4.2.3.5 Function Code FC5 (Write Coil)

This function writes a single output bit to the slave device.

Request

The request specifies the reference number (output address) of output bit to be written. The reference number of the request is zero based; therefore, the first coil starts at address 0.

Example: Turn ON the second output bit (address 1):

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

4.2.3.6 Function Code FC6 (Write single register)

This function writes the value of one single output register to a slave device in word format.

Request

The request specifies the reference number (register address) of the first output word to be written. The value to be written is specified in the “Register Value” field. The reference number of the request is zero based; therefore, the first register starts at address 0.

Example: Write a value of 0x1234 to the second output register.

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

4.2.3.7 Function Code FC11 (Get comm event counter)

This function returns a status word and an event counter from the slave device’s communication event counter. By reading the current count before and after a series of messages, a master can determine whether the messages were handled normally by the slave.

Following each successful new processing, the counter counts up. This counting process is not performed in the case of exception replies, poll commands or counter inquiries.

Request

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

4.2.3.8 Function Code FC15 (Force Multiple Coils)

This function sets a sequence of output bits to 1 or 0 in a slave device. The maximum number is 256 bits.

Request

The request message specifies the reference number (first coil in the sequence), the bit count (number of bits to be written), and the output data. The output coils are zero-based; therefore, the first output point is 0.

In this example 16 bits are set, starting with the address 0. The request contains 2 bytes with the value 0xA5F0, or 1010 0101 1111 0000 in binary format.

The first data byte transmits the value of 0xA5 to the addresses 7 to 0, whereby 0 is the lowest value bit. The next byte transmits 0xF0 to the addresses 15 to 8, whereby the lowest value bit is 8.

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

4.2.3.9 Function Code FC16 (Write multiple registers)

This function writes a sequence of registers in a slave device in word format.

Request

The Request specifies the reference number (starting register), the word count (number of registers to write), and the register data . The data is sent as 2 bytes per register. The registers are zero-based; therefore, the first output is at address 0.

Example: Set data in registers 0 and 1:

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

4.2.3.10 Function Code FC22 (Mask Write Register)

This function manipulates individual bits within a register using a combination of an AND mask, an OR mask, and the register's current content.
Request

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

4.2.3.11 Function Code FC23 (Read/Write multiple registers)

This function performs a combination of a read and write operation in a single request. The function can write the new data to a group registers, and then return the data of a different group.

Request

The reference numbers (addresses) are zero-based in the request message; therefore, the first register is at address 0.
The request message specifies the registers to read and write. The data is sent as 2 bytes per register.

Example: The data in register 3 is set to value 0x0123, and values 0x0004 and 0x5678 are read out of the two registers 0 and 1.

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

Note
If register areas for read and write overlap, the results are undefined.

4.2.4 MODBUS Register Mapping

The following tables display the MODBUS addressing and the corresponding IEC61131 addressing for the process image, the PFC variables, the 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]	Modbus-Address [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 4.2.5.3 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]	Modbus-Address [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 4.2.5.3 Configuration Functions)

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 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		Memory Range	Description
[dec]	[hex]		
0 ... 511	0x0000 ... 0x01FF	Physical Input Area (1)	First 512 digital inputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area (1)	First 512 digital outputs
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: "Illegal data address"
4096 ... 8191	0x1000 ... 0x1FFF	%QX256.0 ...%QX511.15	PFC-OUT-Area Volatile PFC Output variables
8192 ... 12287	0x2000 ... 0x2FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Volatile PFC Input variables
12288 ... 32767	0x3000 ... 0x7FFF	%MW0 ... %MW1279.15	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		Memory Range	Description
[dec]	[hex]		
0 ... 511	0x0000 ... 0x01FF	Physical Output Area (1)	First 512 digital outputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area (1)	First 512 digital outputs

1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 8191	0x1000 ... 0x1FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Volatile PFC Input variables
8192 ... 12287	0x2000 ... 0x2FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Volatile PFC Input variables
12288 ... 32767	0x3000 ... 0x7FFF	%MW0 ... %MW1279.15	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”

4.2.5 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
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

4.2.5.1 Description of the internal variables

4.2.5.1.1 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)

4.2.5.1.2 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) ($2^{(\text{Funtion code-1})} + 2^{(\text{Funtion code-1...})}$). 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 occured.

Register address 0x1005 (MODBUS Address 404102)	
Designation	Watchdog stoppen, 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. Proceedure: 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 Adresse 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.

4.2.5.2 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 definitons)

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 definitons)

4.2.5.3 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 a word count of upto 9)	
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 2 words BadFunction;M -> Invalid function code 2 words Bad Address; -> Invalid register address 2 words BadData; -> Invalid value 2 words TooManyRegisters; -> Number of the registers which can be worked on is too large, Read/Write 125/100 2 words TooManyBits -> Number of the coils which can be worked on is too large, Read/Write 2000/800 2 words 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 buscoupler will allow a ModbusTCP 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 upto 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 upto 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 upto 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 upto 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

4.2.5.4 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 upto 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 upto 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 upto 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 upto 32)	
Value	Description, INFO_LOADER_INFO
Access	read
Description	Information to the programming of the firmware, 32 words

4.2.5.5 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 apears, 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 positiv number, GP_MAX_POS
Access	Read
Description	Constant in order to control arithmetic.

Register address 0x2006 (MODBUS Address 408199)	
Value	Maximum negativ number, GP_MAX_NEG
Access	Read
Description	Constant in order to control arithmetic.

Register address 0x2007 (MODBUS Address 408200)	
Value	Maximum half positiv number, GP_HALF_POS
Access	Read
Description	Constant in order to control arithmetic.

Register address 0x2008 (MODBUS Address 408201)	
Value	Maximum half negativ 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.

4.3 EtherNet/IP (Ethernet/Industrial Protocol)

4.3.1 General

EtherNet/IP stands for Ethernet Industrial Protocol and defines an open industry standard that extends the classic Ethernet with an industrial protocol. This standard was jointly developed by ControlNet International (CI) and the Open DeviceNet Vendor Association (ODVA) with the help of the Industrial Ethernet Association (IEA).

This communication system enables devices to exchange time-critical application data in an industrial environment. The spectrum of devices ranges from simple I/O devices (e.g., sensors) through to complex controllers (e.g., robots).

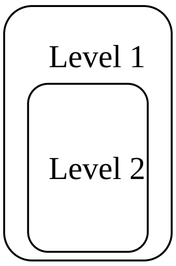
EtherNet/IP is based on the TCP/IP protocol family and consequently uses the bottom 4 layers of the OSI layer model in unaltered form so that all standard Ethernet communication modules such as PC interface cards, cables, connectors, hubs and switches can also be used with EtherNet/IP.
Positioned above the transport layer is the encapsulation protocol, which enables use of the Control & Information Protocol (CIP) on TCP/IP and UDP/IP.

CIP, as a major network independent standard, is already used with ControlNet and DeviceNet. Therefore, converting from one of these protocols to EtherNet/IP is easy to do.
Data exchange takes place with the help of an object model.
In this way, ControlNet, DeviceNet and EtherNet/IP have the same application protocol and can therefore jointly use device profiles and object libraries. These objects enable plug-and-play interoperability between complex devices of different manufacturers.

In order to clarify the interrelationships between DeviceNet, ControlNet and EtherNet/IP, the following diagram presents the associated ISO/OSI reference model.

<table>
<tr><td colspan="4">Application device profiles
(e.g. positioning controllers, semi-conductors, pneumatic valves)</td><td rowspan="4">CIP</td></tr>
<tr><td>7 Application layer</td><td colspan="3">CIP application object library</td></tr>
<tr><td>6 Presentation layer</td><td colspan="3">CIP data management services
(explicit messages, I/O messages)</td></tr>
<tr><td>5 Session layer</td><td colspan="3">CIP message routing, connection management</td></tr>
<tr><td>4 Transport layer</td><td rowspan="3">DeviceNet or ControlNet transport
(transmission control, addressing)</td><td colspan="2">Encapsulation protocol</td><td rowspan="5">ETHERNET/IP</td></tr>
<tr><td rowspan="2">3 Network layer</td><td>TCP</td><td>UDP</td></tr>
<tr><td colspan="2">IP</td></tr>
<tr><td>2 Data Link layer</td><td>CAN (CSMA/NBA) or ControlNet (CTDMA)</td><td colspan="2">Ethernet
(CSMA/CD)</td></tr>
<tr><td>1 Physical layer</td><td>DeviceNet or ControlNet physical interface</td><td colspan="2">Ethernet physical interface</td></tr>
</table>

4.3.2 Characteristics of the EtherNet/IP Protocol Software



Level 1: Explicit Messages Server

Level 2: Level 1 + I/O Messages Server

- Unconnected Message Manager (UCMM) Client and Server
- 128 Encapsulation Protocol sessions
- 128 Class 3 or Class 1 Connections combined

  Class 3 connection – explicit messages (connection oriented, client and server)
  Class 1 connection – I/O messages (connection oriented, client and server)

4.3.3 Object model

4.3.3.1 General

For network communication, EtherNet/IP uses an object model, in which are described all of the functions and data of a device.
Each node in the network is represented as a collection of objects.

A number of terms relating to object models are defined below:

- **Object:**
  The object model consists of classes of objects. An object is an abstract representation of individual related components within a device. It is defined by its data or attributes, the functions or services it provides externally and by its defined behaviour.

- **Class:**
  A class contains related components (objects) of a product, organized in instances.

- **Instance:**
  An instance consists of different variables (attributes) that describe the properties of this instance. Different instances of a class have the same services, the same behaviour and the same variables (attributes).
  They can, however, have different variable values.

- **Variable (attribute):**
  The variables (attributes) represent the data a device provides over EtherNet/IP. These include the current values of, for example, a configuration or an input. Typical attributes are configuration or status information.

- **Service:**
  Services are used to access classes or the attributes of a class or to generate specific events. These services execute defined actions such as the reading of variables or the resetting of a class. For each class, there exists a fixed set of services.

- **Behaviour:**
  The behaviour defines how a device reacts as a result of external events such as changed process data or internal events such as lapsing timers.

4.3.3.2 Classes

The following classes are supported by the EtherNet/IP software:

4.3.3.2.1 CIP Common Classes

Class	**Name**
01_{hex}	Identity
02_{hex}	Message Router
04_{hex}	Assembly
05_{hex}	Connection
06_{hex}	Connection Manager
$F5_{hex}$	TCP/IP Interface Object
$F6_{hex}$	Ethernet Link Object

4.3.3.2.2 WAGO specific Classes

Class	**Name**
64_{hex}	Controller configuration Object
65_{hex}	Discrete Input Point
66_{hex}	Discrete Output Point
67_{hex}	Analog Input Point
68_{hex}	Analog Output Point
69_{hex}	Discrete Input Point Extended 1
$6A_{hex}$	Discrete Output Point Extended 1
$6B_{hex}$	Analog Input Point Extended 1
$6C_{hex}$	Analog Output Point Extended 1
$6D_{hex}$	Discrete Input Point Extended 2
$6E_{hex}$	Discrete Output Point Extended 2
$6F_{hex}$	Analog Input Point Extended 2
70_{hex}	Analog Output Point Extended 2
71_{hex}	Discrete Input Point Extended 3
72_{hex}	Discrete Output Point Extended 3
73_{hex}	Analog Input Point Extended 3
74_{hex}	Analog Output Point Extended 3
80_{hex}	Module configuration
81_{hex}	Module configuration Extended 1
$A0_{hex}$	Input fieldbus variable USINT
$A1_{hex}$	Input fieldbus variable USINT Extended 1
$A2_{hex}$	Input fieldbus variable USINT Extended 2

Class	Name
A3 hex	Output fieldbus variable USINT
A4 hex	Output fieldbus variable USINT Extended 1
A5 hex	Output fieldbus variable USINT Extended 2
A6 hex	Input fieldbus variable UINT
A7 hex	Input fieldbus variable UINT Extended 1
A8 hex	Output fieldbus variable UINT
A9 hex	Output fieldbus variable UINT Extended 1
AA hex	Input fieldbus variable UDINT
AB hex	Input fieldbus variable UDINT Offset UINT
AC hex	Output fieldbus variable UDINT
AD hex	Output fieldbus variable UDINT Offset UINT

4.3.3.2.3 Explanations of the Object Description

Attribute ID:	Integer value which is assigned to the corresponded attribute
Access:	Set The attribute can be accessed by means of Set_Attribute services. Important: All the set attributes can also be accessed by means of Get_Attribute services. Get The attribute can be accessed by means of Get_Attribute services
NV:	NV (non volatile) The attribute is permanently stored in the controller. V (volatile) The attribute is not permanently stored in the controller. Note: If this column is missing, all attributes have the type V
Name:	Designation of the attribute
Data type:	Designation of the CIP data type of the attribute
Description:	Short description for the Attribute
Default value:	Factory settings

4.3.3.2.4 Identity (01 hex)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Maximum Instance	0x0001
3	Get	Max ID Number of Class Attributes	UINT	Maximum number of Class attributes	0x0000
4	Get	Max ID Number of Instance Attribute	UINT	Maximum number of instance attributes	0x0000

Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Vendor ID	UINT	Manufacturer identification	40 (0x0028)
2	Get	Device Type	UINT	General type designation of the product	12 (0x000C)
3	Get	Product Code	UINT	Designation of the controller	e.g. 841 (0x0349)
4	Get	Revision	STRUCT of:	Revision of the Identity Objects	Depending on the firmware
		Major Revision	USINT		
		Minor Revision	USINT		
5	Get	Status	WORD	Current status of the device	Bit 0: Assignment to a master Bit 1=0 (reserved) Bit 2: Configured: (=0: Configuration is unchanged; =1: Configuration is different to the manufacturers parameters) Bit 3=0 (reserved) Bit 4-7: Extended Device Status: (=0010: at least one faulted I/O connection, =0011: no I/O connection established) Bit 8-11: not used Bit 12-15=0 (reserved)
6	Get	Serial Number	UDINT	Serial number	The last 4 digits of MAC ID
7	Get	Product Name	SHORT_STRING	Product name	e.g. “WAGO Ethernet (10/100 Mbps)-FBC

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
01 hex	yes	yes	Get_Attribute_All	Supplies contents of all attributes
05 hex	no	yes	Reset	Implements the reset service. Service parameter: 0: Emulates a Power On reset 1: Emulates a Power On reset and re-establishes factory settings
0E hex	no	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

4.3.3.2.5 Message Router (02 hex)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Number of Attributes	UINT	Number of attributes	0 (0x0000)
3	Get	Number of Services	UINT	Number of services	0 (0x0000)
4	Get	Max ID Number of Class Attributes	UINT	Maximum number of class attributes	0 (0x0000)
5	Get	Max ID Number of Instance Attributes	UINT	Maximum number of instance attributes	0 (0x0000)

Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	ObjectList	STRUCT of:		
		Number	UINT		40
		Classes	UINT		01 02 04 00 06 00 F4 00 F5 00 F6 00 64 00 65 0066 0067 00 68 00 69 00 6A 00 6B 00 6C 00 6D 00 6E 00 6F 00 70 00 71 00 72 00 73 00 74 00 80 00 81 00 A0
2	Get	NumberAvailable	UINT	Varable	0x80

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
01 hex	yes	no	Get_Attribute_All	Supplies contents of all attributes
0E hex	no	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

4.3.3.2.6 Assembly (04_{hex})

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	2 (0x0002)

4.3.3.2.6.1 Static Assembly Instances

Instance 101 (65_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Set	Data	ARRAY of BYTE	Reference on the process image: analog and digital output data	-

Instance 102 (66_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Set	Data	ARRAY of BYTE	Reference on the process image: only digital output data	-

Instance 103 (67_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Set	Data	ARRAY of BYTE	Reference of the process image: only analog output data	-

Instance 104 (68_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Reference of the process image: analog and digital input data + Status	-

Instance 105 (69_{hex})

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Reference of the process image: only digital input data + Status	-

Instance 106 (6A $_{hex}$)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Reference of the process image: only analog input data + Status	-

Instance 107 (6B $_{hex}$)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Reference of the process image: analog and digital input data	-

Instance 108 (6C $_{hex}$)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Reference of the process image: only digital input data	-

Instance 109 (6D $_{hex}$)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Reference of the process image: only analog input data	-

Instance 110 (6E $_{hex}$)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Reference of the process image: only PFC output variables	-

Instance 111 (6F $_{hex}$)

Attribute ID	Access	Name	Data type	Description	Default value
3	Set	Data	ARRAY of BYTE	Reference of the process image: only PFC input variables	-

Instance 198 (C6 $_{hex}$) „Input Only“

This instance is used to establish a connection when no outputs are to be addressed or when inputs, which are already being used in an exclusive owner connection, are to be interrogated . The data length of this instance is always zero.
This instance can only be used in the "consumed path" (seen from the slave device).

Instance 199 (C7 $_{hex}$)

This instance is used to establish a connection based on an existing exclusive owner connection. The new connection also has the same transmission parameters as the exclusive owner connection. When the exclusive owner connection is cleared, this connection, too, is automatically cleared. The data length of this instance is always zero.
This instance can only be used in the "consumed path" (from the point of view of the slave device).

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	no	yes	Set_Attribute_Single	Modifies an attribute value

4.3.3.2.7 Port Class (F4 $_{hex}$)

Class

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

Instance 1

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Port Type	UINT	-	0x0004
2	Get	V	Port Number	UINT	Cip Port number	0x0002 (EtherNet/IP)
3	Get	V	Port Object	UINT	Number of 16 bit words in the following path	0x0002
				Padded EPATH	Object, which manages this port	0x20 0xF5 0x24 0x01
4	Get	V	Port Name	Short String	Portname	0x00
7	Get	V	Node Address	Padded EPATH	Port segment (IP address)	-

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
01 hex	yes	yes	Get_Attribute_All	Supplies contents of all attributes
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

4.3.3.2.8 TCP/IP Interface (F5 $_{hex}$)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	
3	Get	Num Instances	UINT	Number of the current instanced connections	

Instance 1

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Status	DWORD	Interface state	-
2	Get	V	Configuration Capability	DWORD	Inferface flags for possible kinds of configuration	0x00000007
3	Set	NV	Configuration Control	DWORD	Specifies, how the device gets ist TCP/IP configuration after the first Power On	0x00000011
4	Get	V	Physical Link Object	STRUCT of		
			Path size	UINT	Size of the path	0x0004
			Path	UINT	Number of 16 Bit words in the following path	0x0002
				Padded EPATH	Logical path, which points to the physical Link object	0x20 0xF6 0x24 0x01
5	Get	NV	Interface Configuration	STRUCT of		
			IP Address	UDINT	IP address	0
			Network Mask	UDINT	Net work mask	0
			Gateway Address	UDINT	IP address of default gateway	0
			Name Server	UDINT	IP address of the primary name of the server	0
			Name Server 2	UDINT	IP address of the secundary name of the server	0
			Domain Name	STRING	Default Domain name	“”
6	Set	NV	Host Name	STRING	Device name	local host

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
01 hex	yes	yes	Get_Attribute_All	Supplies contents of all attributes
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	no	yes	Set_Attribute_Single	Modifies an attribute value

4.3.3.2.9 Ethernet Link (F6 hex)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	2 (0x0002)
2	Get	Max Instance	UDINT	Max. number of instances	0x0001
3	Get	Num Instances	UDINT	Number of the current instanced connections	-

Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Interface Speed	UDINT	Transfer rate	10 (0x0A) or 100 (0x64)
2	Get	Interface Flags	DWORD	Interface configuration and status information	Bit 0: Link active Bit 1: Full Duplex
3	Get	Physical Address	ARRAY of 6 UINTs	MAC layer address	MAC ID of the device

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
01 hex	yes	yes	Get_Attribute_All	Supplies contents of all attributes
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

4.3.3.2.10 Controller Configuration (64 hex)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	1 (0x0001)

Instance 1

Attribute ID	Access	NV	Name	Data type	Description	Default value
5 (0x05)	Get	V	ProcessState	USINT	State of controller Error mask Bit 0 Internal bus error Bit 3 Module diagnostics (0x08) Bit 7 Fieldbus error (0x80)	0
6 (0x06)	Get	V	DNS_i_Trm	UINT	Module diagnostics	0

Attribute ID	Access	NV	Name	Data type	Description	Default value
			nldia		Bit 0..7:Module number Bit 8..14:Module channel Bit 15: 0/1 Error repair/arisen	
7 (0x07)	Get	V	CnfLen.AnalogOut	UINT	Number of I/O bits for the analog outputs	-
8 (0x08)	Get	V	CnfLen.AnalogInp	UINT	Number of I/O bits for the analog inputs	-
9 (0x09)	Get	V	CnfLen.DigitalOut	UINT	Number of I/O bits for the digital outputs	-
10 (0x0A)	Get	V	CnfLen.DigitalInp	UINT	Number of I/O bits for the digital inputs	-
11 (0x0B)	Set	NV	Bk_Fault_Reaction	USINT	Fieldbus error reaction 0: stop local I/O cycles 1: set all output to 0 2: no error reaction 3: no error reaction 4: PFC task takes over control of the outputs	1
12..26 (0x0C...0x1A)	Reserved for compatibility to DeviceNet					
40..43 (0x28... 0x2B)	Reserved for compatibility to DeviceNet					
45 (0x2D)	Get	V	Bk_Led_Err_Code	UINT	I/O LED Error Code	0
46 (0x2E)	Get	V	Bk_Led_Err_Arg	UINT	I/O LED Error Argument	0
100 (0x64)	Set	NV	Bk_FbInp_Var_Cnt	UINT	Determines the number of bytes for the PFC input fieldbus variables, which are added to the Assembly object. This number is added to the consuming path. Assembly Instances (101..103)	0
101 (0x65)	Set	NV	Bk_FbOut_Var_Cnt	UINT	Determines the number of bytes for the PFC output fieldbus variables, which are added to the Assembly object. This number is added to the producing path. Assembly Instances (104..109)	0
102 (0x66)	Set	NV	Bk_FbInp_PlcOnly_Var_Cnt	UINT	Determines the number of bytes for the PFC input fieldbus variables, which are received via Assembly instance 111.	4
103 (0x67)	Set	NV	Bk_FbInp_StartPlc_Var_Cnt	UINT	Determines starting from which position the PFC input fieldbus variables for the assembly instance 111 to be received.	0
104 (0x68)	Set	NV	Bk_FbOut_PlcOnly_Var_Cnt	UINT	Determines the number of bytes for the PFC output fieldbus variables, which are received via Assembly instance 110.	4
105 (0x69)	Set	NV	Bk_FbOut_StartPlc_Var_Cnt	UINT	Determines starting from which position the PFC output fieldbus variables for the assembly instance 110 to be received.	0

Attribute ID	Access	NV	Name	Data type	Description	Default value
120 (0x78)	Set	NV	Bk_HeaderCfgOT	UINT	Indicates whether the RUN/IDLE header is used Originator -> Target direction 0 is used 1 is not used	0x0000
121 (0x79)	Set	NV	Bk_HeaderCfgTO	UINT	Indicates whether the RUN/IDLE header is used Originator -> Target direction 0 is used 1 is not used	0x0001

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	no	yes	Set_Attribute_Single	Modifies an attribute value

4.3.3.2.11 Discrete Input Point (65 $_{hex}$)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 1 ... 255 (1. to 255. Digital Input Value)

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	DipObj_Value	BYTE	Digital input (only Bit 0 is valid)	-

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

4.3.3.2.12 Discrete Output Point (66 hex)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 1..255 (1. to 255. Digital Output Value)

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	DopObj_Value	BYTE	Digital output (only Bit 0 is valid)	-

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_ Single	Supplies contents of the appropriate attribute
10 hex	no	yes	Set_Attribute_ Single	Modifies an attribute value

4.3.3.2.13 Analog Input Point (67 hex)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 1..255 (1. to 255. Analog input value)

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	Array of Byte	Analog input	-
2	Get	AipObj_Value_Lengt h	USINT	Length of the input data AipObj_Value (in byte)	-

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_ Single	Supplies contents of the appropriate attribute

4.3.3.2.14 Analog Output Point (68 hex)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 1 ... 255 (1. to 255. Analog output value)

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AopObj_Value	Array of Byte	Analog output	-
2	Get	AopObj_Value_Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	no	yes	Set_Attribute_Single	Modifies an attribute value

4.3.3.2.15 Discrete Input Point Extended 1..3 (69 hex, 6D hex, 71 hex)

Same as the Discret Input Point (65 hex), however it contains the extended digital inputs:

69 hex : Digital Input 256 ..510
6D hex : Digital Input 511 ..765
71 hex : Digital Input 766 ..1020

4.3.3.2.16 Discrete Output Point Extended 1..3 (6A hex, 6E hex, 72 hex)

Same as the Discret Input Point (66 hex), however it contains the extended digital inputs:

6A hex : Digital Output 256 ..510
6E hex : Digital Output 511 ..765
72 hex : Digital Output 766 ..1020

4.3.3.2.17 Analog Input Point Extended 1..3 (6B hex, 6F hex, 73 hex)

Same as the Analog Input Point Class (67 hex), however it contains the extended analog inputs:

6B hex : Analog Inputs 256 ..510
6F hex : Analog Inputs 511 ..765
73 hex : Analog Inputs 766 ..1020

4.3.3.2.18 Analog Output Point Extended 1..3 (6C hex, 70 hex, 74 hex)

Same as the Analog Input Point Class (68 hex), however it contains the extended analog outputs:

6C hex : Analog Outputs 256 ..510

70 hex : Analog Outputs 511 ..765

74 hex : Analog Outputs 766 ..1020

4.3.3.2.19 Module configuration (80 hex)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	-

Instance 1..255 (0. to 254. Modul)

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	ModulDescription	WORD	Description of the connected modules (module 0 = controller) Bit 0: Module has inputs Bit 1: Module has outputs Bit 8-14: Internal data width in bit Bit 15: 0/1 Analog/digital module At analog modules designate bits 0-14 the type of the module displayed (e.g., 401 for the module 750-401).	-

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_ Single	Supplies contents of the appropriate attribute

4.3.3.2.20 Module configuration Extended (81 hex)

Same as the Module Configuration Class (80 hex), however this class only contains the description of module 255.

4.3.3.2.21 Input fieldbus variable USINT (A0 hex)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x0FF)

Instance 1..255 (1. to 255. Input variable)

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_In_Var	USINT	Fieldbus Input variable of the PLC	0

This instance corresponds to the PFC's input variables located at address %IB2552-%IB2807 in a WAGO-I/O-PRO application.

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	no	yes	Set_Attribute_Single	Modifies an attribute value

4.3.3.2.22 Input fieldbus variable USINT Extended 1 ($A1_{hex}$)

Same as the Input Fieldbus Variable USINT Class ($A0_{hex}$), however this contains the PLC input variables 256..510.
This instance corresponds to the PFC's input variables located at address %IB2808-%IB3061 in a WAGO-I/O-PRO application.

4.3.3.2.23 Input fieldbus variable USINT Extended 2 ($A2_{hex}$)

Same as the Input Fieldbus Variable USINT Class ($A0_{hex}$), however this contains the PLC input variables 511..512.
This instance corresponds to the PFC's input variables located at address %IB3062-%IB3063 in a WAGO-I/O-PRO application.

Class attribute max. Instance = 2

4.3.3.2.24 Output fieldbus variable USINT ($A3_{hex}$)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x0FF)

Instance 1..255 (1. to 255. Output variable)

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Fb_Out_Var	USINT	Fieldbus Output variable of the PLC	0

This instance corresponds to the PFC's output variables located at address %QB2552-%QB2807 in a WAGO-I/O-PRO application.

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

4.3.3.2.25 Output fieldbus variable USINT Extended 1 (A4 $_{hex}$)

Same as the Output Fieldbus Variable USINT Class (A3 $_{hex}$), however this contains the PLC output variables 256..510.
This instance corresponds to the PFC’s output variables located at address %QB2808-%QB3062 in a WAGO-I/O-PRO application.

4.3.3.2.26 Output fieldbus variable USINT Extended 2 (A5 $_{hex}$)

Same as the Output Fieldbus Variable USINT Class (A3 $_{hex}$), however this contains the PLC output variables 511..512.
This instance corresponds to the PFC’s output variables located at address %QB3063-%QB3064 in a WAGO-I/O-PRO application.

Class attribute max. Instance = 2

4.3.3.2.27 Input fieldbus variable UINT (A6 $_{hex}$)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x0FF)

Instance 1..255 (1. to 255. Input Variable)

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_In_Var	UINT	Fieldbus Input variable of the PLC	0

This instance corresponds to the PFC’s input variables located at address %IW1276-%IW1530 in a WAGO-I/O-PRO application.

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	no	yes	Set_Attribute_Single	Modifies an attribute value

4.3.3.2.28 Input fieldbus variable USINT Extended 1 (A7 $_{hex}$)

Same as the Input Fieldbus Variable UINT Class (A6 $_{hex}$), however this contains only the PLC input variable 256.
This instance corresponds to the PFC’s input variable located at address %IW1531 in a WAGO-I/O-PRO application.

Class attribute max. Instance = 1

4.3.3.2.29 Output fieldbus variable UINT (A8 $_{hex}$)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x0FF)

Instance 1..255 (1. to 255. Output variable)

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Fb_Out_Var	UINT	Fieldbus Output variable of the PLC	0

This instance corresponds to the PFC's output variables located at address %QW1276-%QW1530 in a WAGO-I/O-PRO application.

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

4.3.3.2.30 Output fieldbus variable UINT Extended 1 (A9 $_{hex}$)

Same as the Output Fieldbus Variable UINT Class (A8 $_{hex}$), however this contains only the PLC output variable 256.
This instance corresponds to the PFC's output variable located at address %QW1531 in a WAGO-I/O-PRO application.

Class attribute max. Instance = 1

4.3.3.2.31 Input fieldbus variable UDINT (AA $_{hex}$)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	128 (0x080)

Instance 1..128 (1. to 128. Input variable)

Attribute ID	Access	Name	Data type	Description	Default value
1	Set	Fb_In_Var	UDINT	Fieldbus Input variable of the PLC	0

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	no	yes	Set_Attribute_Single	Modifies an attribute value

4.3.3.2.32 Input fieldbus variable UDINT Offset (AB $_{hex}$)

Same as the Input Fieldbus Variable UDINT (AA $_{hex}$), however it has an offset of 2 bytes.

4.3.3.2.33 Output fieldbus variable UDINT (AC $_{hex}$)

Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	128 (0x080)

Instance 1..128 (1. to 128. Output variable)

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Fb_Out_Var	UDINT	Fieldbus output variable of the PLC	0

Common Services

Service code	Service available		Service Name	Description
	Class	Instance		
0E hex	yes	yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

4.3.3.2.34 Output fieldbus variable UDINT Offset (AD $_{hex}$)

Same as the Output fieldbus Variable UDINT (AC $_{hex}$), however with an offset of 2 bytes.

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 on the web pages:
www.wago.com → Service → Download → Documentation.

More Information
Current information on the modular WAGO-I/O-SYSTEM is available in the Internet under:
www.wago.com

5.1.1 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; diagnostic
750-419	2 Channel, DC 24 V, 3.0 ms, 2- to 3-conductor connection; high-side switching; diagnostic
750-421, 753-421	2 Channel, DC 24 V, 3.0 ms, 2- to 3-conductor connection; high-side switching; diagnostic
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; lowside 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

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; with diagnostics
750-507, 753-507	2 Channel, DC 24 V, 2.0 A, short-circuit-protected; high-side switching; with diagnostics; No longer available, replaced by 750-508
750-508	2 Channel, DC 24 V, 2.0 A, short-circuit-protected; high-side switching; with diagnostics; Replacement for 750-508
750-535	2 Channel, DC 24 V, EEx i, short-circuit-protected; PNP-positive 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
750-516, 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; with 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, isolated output, 1 make contact, bistable, manual operation
750-514, 753-514	2 Channel, AC 125 V , AC 0.5 A , DC 30 V, DC 1 A, isolated outputs, 2 changeover contacts
750-517, 753-517	2 Channel, AC 230 V, 1 A, isolated outputs, 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, isolated outputs, 2 make contacts

5.1.3 Analog Intput Modules

AI 0 - 20 mA	
750-452, 753-452	2 Channel, 0 - 20 mA, Differential Inputs
750-465, 753-465	2 Channel, 0 - 20 mA, single-ended (S.E.)
750-472, 753-472	2-channel, 0 - 20 mA, 16 Bit, single-ended (S.E.)
750-480	2-channel, 0 - 20 mA ,Differential Inputs
750-453, 753-453	4 Channel, 0 - 20 mA, single-ended (S.E.)
AI 4 - 20 mA	
750-454, 753-454	2 Channel, 4 - 20 mA,Differential Inputs
750-474, 753-474	2 Channel, 4 - 20 mA, 16 Bit, single-ended (S.E.)
750-466, 753-466	2 Channel, 4 - 20 mA, single ended (S.E.)
750-485	2 Channel, 4 - 20 mA, EEx i, single ended (S.E.)
750-492, 753-492	2 Channel, 4 - 20 mA, Isolated Differential Inputs
750-455, 753-455	4 Channel, 4 - 20 mA, single ended (S.E.)
AI 0 - 1 A	
750-475, 753-475	2-channel, 0 - 1 A AC/DC ,Differential Inputs
AI 0 - 5 A	
750-475/020-000, 753-475/020-000	2-channel, 0 - 5 A AC/DC ,Differential Inputs
AI 0 - 10 V	
750-467, 753-467	2 Channel, DC 0 - 10 V, single-ended (S.E.)
750-477, 753-477	2 Channel, AC/DC 0 - 10 V,Differential Inputs
750-478, 753-478	2 Channel, DC 0 - 10 V, single-ended (S.E.)
750-459, 753-459	4 Channel, DC 0 - 10 V, single-ended (S.E.)
750-468	4 Channel, DC 0 - 10 V, single-ended (S.E.)
AI DC ± 10 V	
750-456, 753-456	2 Channel, DC ± 10 V,Differential Inputs
750-479, 753-479	2 Channel, DC ± 10 V,Differential Measurement Input
750-476, 753-476	2 Channel, DC ± 10 V, single-ended (S.E.)
750-457, 753-457	4 Channel, DC ± 10 V, single-ended (S.E.)
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 with diagnostics Sensor types: J, K, B, E, N, R, S, T, U
750-469, 753-469	2 Channel, thermocouples with diagnostics 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

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

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/ 16Bit / 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 Encor Interface, TTL level squarewave
750-634	Incremental Encor Interface, DC 24 V
750-637	Incremental Encor Interface RS 422, cam outputs
750-635, 753-635	Digital Pulse Interface
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
750-665/000-001	4FDO 0.5A / 4FDI 24V DC PROFIsafe
750-666/000-001	1FDO 10A / 2FDO 0.5A / 2FDI 24V PROFIsafe
RTC Module	
750-640	RTC Module

5.1.6 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, passiv
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, passiv
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 for field side power supply
750-626	Filter Module for 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 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 bi-directional 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.

<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>D0</td><td>C/S</td><td>Data byte</td><td>Control/Status byte</td></tr>
<tr><td>1</td><td>D2</td><td>D1</td><td colspan="2" rowspan="2">Data bytes</td></tr>
<tr><td>2</td><td>D4</td><td>D3</td></tr>
</table>

Data Exchange Module

750-654, (and the variation /000-001)

The Data Exchange modules have a total of 4 bytes of user data in both the Input and Output Process Image. The following tables illustrate the Input and Output Process Image, which has a total of 2 words mapped into each image. Word alignment is applied.

<table>
<tr><th colspan="4">Input and Output Process Image</th></tr>
<tr><th rowspan="2">Offset</th><th colspan="2">Byte Destination</th><th rowspan="2">Remark</th></tr>
<tr><th>High Byte</th><th>Low Byte</th></tr>
<tr><td>0</td><td>D1</td><td>D0</td><td rowspan="2">Data bytes</td></tr>
<tr><td>1</td><td>D3</td><td>D2</td></tr>
</table>

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 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.

<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>C1/S1</td><td>C0/S0</td><td>extended Control/Status byte</td><td>Control/Status byte</td></tr>
<tr><td>1</td><td>D1</td><td>D0</td><td colspan="2" rowspan="3">Data bytes</td></tr>
<tr><td>2</td><td>D3</td><td>D2</td></tr>
<tr><td>3</td><td>D5</td><td>D4</td></tr>
</table>

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 suppressable mailbox (Mode 2), the mailbox and the cyclical process data are mapped next.
The following words contain the remaining process data.

Input and Output Process Image				
Offset	Byte Destination		Remark	
	High Byte	Low Byte		
0	-	C0/S0	not used	Control/Status byte
1	D1	D0	Mailbox (0, 3, 5, 6 or 9 words) / Process data (0-16 words)	
2	D3	D2		
3	D5	D4		
...	...	...		
max. 23	D45	D44		

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

5.3 Process Data Architecture for EtherNet/IP

With some I/O modules, the structure of the process data is fieldbus specific.

In the case of a coupler/controller with EtherNet/IP, the process image uses a word structure (with word alignment). The internal mapping method for data greater than one byte conforms to the Intel format.

The following section describes the process image for various WAGO-I/O-SYSTEM 750 and 753 I/O modules when using a coupler/controller with EtherNet/IP.

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 the structure of the process data mapping identical.

5.3.1 Digital Input Modules

Digital input modules supply one bit of data per channel to specify the signal state for the corresponding channel. These bits are mapped into the Input Process Image.

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).

Each input channel seizes one Instance in the Discrete Input Point Object (Class 0x65).

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

The input modules seize 2 Instances in Class (0x65).

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

The input modules seize 2 Instances in Class (0x65).

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

The input modules seize 4 Instances in Class (0x65).

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

The input modules seize 4 Instances in Class (0x65).

Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Acknowledgement bit Q 2 Channel 2	Acknowledgement bit Q 1 Channel 1	0	0

And the input modules seize 4 Instances in Class (0x66).

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

The input modules seize 4 Instances in Class (0x65).

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

The input modules seize 8 Instances in Class (0x65).

5.3.2 Digital Output Modules

Digital output modules use one bit of data per channel to control the output of the corresponding channel. These bits are mapped into the Output Process Image.

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.

Each output channel seizes one Instance in the Discrete Output Point Object (Class 0x66).

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“

The output modules seize 2 Instances in Class (0x65).

Output Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
						not used	controls DO 1 Channel 1

And the output modules seize 2 Instances in Class (0x66).

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

The output modules seize 2 Instances in Class (0x66).

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

The output modules seize 2 Instances in Class (0x65).

Output Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
						controls DO 2 Channel 2	controls DO 1 Channel 1

And the output modules seize 2 Instances in Class (0x66).

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

The output modules seize 4 Instances in Class (0x65).

Output Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				not used	not used	controls DO 2 Channel 2	controls DO 1 Channel 1

And the output modules seize 4 Instances in Class (0x66).

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

The output modules seize 4 Instances in Class (0x66).

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

The output modules seize 4 Instances in Class (0x65).

Output Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

And the output modules seize 4 Instances in Class (0x66).

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

The output modules seize 8 Instances in Class (0x66).

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

The output modules seize 8 Instances in Class (0x65).

Output Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

And the output modules seize 8 Instances in Class (0x66).

5.3.3 Analog Input Modules

The hardware of an analog input module has 16 bits of measured analog data per channel and 8 bits of control/status. However, the coupler/controller with EtherNet/IP does not have access to the 8 control/status bits. Therefore, the coupler/controller with EtherNet/IP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Input Process Image.
When digital input modules are also present in the node, the analog input data is always mapped into the Input Process Image in front of the digital data.
Each input channel seizes one Instance in the Analog Input Point Object (Class 0x67).

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}

The input modules represent 2x2 bytes and seize 2 Instances in Class (0x67).

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

The input modules represent 2x2 bytes and seize 2 Instances in Class (0x67).

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

The input modules represent 4x2 bytes and seize 4 Instances in Class (0x67).

5.3.4 Analog Output Modules

The hardware of an analog output module has 16 bits of measured analog data per channel and 8 bits of control/status. However, the coupler/controller with EtherNet/IP does not have access to the 8 control/status bits. Therefore, the coupler/controller with EtherNet/IP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Output Process Image.

When digital output modules are also present in the node, the analog output data is always mapped into the Output Process Image in front of the digital data.

Each output channel seizes one Instance in the Analog Output Point Object (Class 0x68).

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

The output modules represent 2x2 bytes and seize 2 Instances in Class (0x68).

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

The output modules represent 4x2 bytes and seize 4 Instances in Class (0x68).

5.3.5 Specialty Modules

WAGO has a host of Specialty I/O modules that perform various functions. With individual modules beside the data bytes also the control/status byte is mapped in the process image. The control/status byte is required for the bi-directional 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.

The Specialty Modules represent as analog modules.
For this, the process input data of the Specialty Modules seize one Instance per channel in the Analog Input Point Object (Class 0x67) and the process output data seize one Instance seize one Instance in the Analog Input Point Object (Class 0x67) per channel in the Analog Output Point Object (Class 0x68).

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	

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

Output Process Image			
Offset	Byte Destination		Remark
	High Byte	Low Byte	
0	-	C	Control byte
1	D1	D0	Counter Setting Value
2	D3	D2	

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-404/000-005

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

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

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

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

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-638, 753-638

The above Counter Modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 2 bytes of control/status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

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

The specialty modules represent 2x3 bytes input data and seize 2 Instances in Class (0x67).

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

And the specialty modules represent 2x3 bytes output data and seize 2 Instances in Class (0x68).

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

The specialty modules represent 2x3 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

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	

The specialty modules represent 2x2 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

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		

The specialty modules represent 1x6 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

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	

The specialty modules represent 2x2 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

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	

The specialty modules represent 2x2 bytes input data and seize 2 Instances in Class (0x67).

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	

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

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	

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-634

The above Incremental Encoder Interface module has 5 bytes of input data (6 bytes in cycle duration measurement mode) and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which has 4 words mapped into each image. Word alignment is applied.

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.

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

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	-	-		

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-637

The above Incremental Encoder Interface Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of encoder data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Input and Output Process Image			
Offset	**Byte Destination**		**Remark**
	High Byte	**Low Byte**	
0	-	C0/S0	Control/Status byte 1
1	D1	D0	Data Value
2	-	C1/S1	Control/Status byte 2
3	D3	D2	Data Value

The specialty modules represent 2x3 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

750-635, 753-635

The above Digital Pulse Interface module has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

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	

The specialty modules represent 1x4 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

RTC Module

750-640

The RTC Module 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				
Instance	Byte Destination		Remark	
	High Byte	Low Byte		
n	ID	C/S	Command byte	Control/Status byte
	D1	D0	Data bytes	
	D3	D2		

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).and seize 1 Instance in Class (0x68).

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

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

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

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

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	

The specialty modules represent 2x2 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

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		

The specialty modules represent 1x8 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

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		
n	-	C0/S0	Not used	Control/Status byte (log. channel 1, Sensor input 1)
	D1	D0	Data bytes (log. channel 1, Sensor input 1)	
n+1	-	C1/S1	Not used	Control/Status byte (log. channel 2, Sensor input 2)
	D3	D2	Data bytes (log. channel 2, Sensor input 2)	
n+2	-	C2/S2	Not used	Control/Status byte (log. channel 3, Sensor input 1)
	D5	D4	Data bytes (log. channel 3, Sensor input 1)	
n+3	-	C3/S3	Not used	Control/Status byte (log. channel 4, Sensor input 2)
	D7	D6	Data bytes (log. channel 4, Sensor input 2)	

The specialty modules represent 4x3 bytes input and output data and seize 4 Instances in Class (0x67) and 4 Instances in Class (0x68).

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 suppressable mailbox (Mode 2), the mailbox and the cyclical process data are mapped next.
The following words contain the remaining process data.

Input and Output Process Image				
Offset	Byte Destination		Remark	
	High Byte	Low Byte		
0	-	C0/S0	not used	Control/Status byte
1	D1	D0	Mailbox (0, 3, 5, 6 or 9 words) / Process data (0-16 words)	
2	D3	D2		
3	D5	D4		
...	...	...		
max. 23	D45	D44		

The specialty modules represent 1x 12...48 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

5.3.6 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

The system modules seize 2 Instances in Class (0x65).

Binary Space Module

750-622

The Binary Space Modules 750-622 behave alternatively like 2 channel digital input modules or output modules and occupy 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

The Binary Space Modules seize 2, 4, 6 or 8 Instances in class (0x65) or in class (0x66)..

6 Application Examples

6.1 Test of MODBUS protocol and fieldbus nodes

You require a MODBUS master to test the function of your fieldbus node. For this purpose, various manufacturers offer a range of PC applications that you can, in part, download from the Internet as free of charge demo versions. One of the programs which is particularly suitable to test your ETHERNET TCP/IP fieldbus node, is for instance **ModScan** from **Win-Tech.**

More information
A free of charge demo version from ModScan32 and further utilities from Win-Tech can be found in the Internet under:
http://www.win-tech.com/html/demos.htm.

ModScan32 is a Windows application that works as a MODBUS master. This program allows you to access the data points of your connected ETHERNET TCP/IP fieldbus node and to proceed with the desired changes.

More information
For a description example relating to the software operation, please refer to:

http://www.win-tech.com/html/modscan32.htm

6.2 Visualization and control using SCADA software

This chapter is intended to give insight into how the WAGO ETHERNET fieldbus coupler/controller can be used for process visualization and control using standard user software.

There is a wide range of process visualization programs, called SCADA Software, from various manufacturers.

More information
For a selection of SCADA products, look under i.e.:
http://www.abpubs.demon.co.uk/scadasites.htm.

SCADA is the abbreviation for Supervisory Control and Data Acquisition.

It is a user-orientated tool used as a production information system in the areas of automation technology, process control and production monitoring.

The use of SCADA systems includes the areas of visualization and monitoring, data access, trend recording, event and alarm processing, process analysis and targeted intervention in a process (control).

The WAGO ETHERNET fieldbus node provides the required process input and output values.

Attention!
When choosing suitable SCADA software, ensure that it provides a MODBUS device driver and supports the MODBUS/TCP functions in the coupler.

Visualization programs with MODBUS device drivers are available from i.e. Wonderware, National Instruments, Think&Do or KEPware Inc., some of which are available on the Internet as demo versions.

The operation of these programs is very specific.
However, a few essential steps are described to illustrate the way an application can be developed using a WAGO ETHERNET fieldbus node and SCADA software in principle.

- The initial prerequisite is that the MODBUS ETHERNET driver has been loaded and MODBUS ETHERNET has been selected.

- Subsequently, the user is requested to enter the IP address for addressing the fieldbus node.
  At this point, some programs allow the user to give the node an alias name, i.e. to call the node "Measuring data". The node can then be addressed with this name.

- Then, a graphic object can be created, such as a switch (digital) or a potentiometer (analog).
  This object is displayed on the work area and is linked to the desired data point on the node.

- This link is created by entering the node address (IP address or alias name) of the desired MODBUS function codes (register/bit read/write) and the MODBUS address of the selected channel.
  Entry is, of course, program specific.
  Depending on the user software the MODBUS addressing of a bus module can be represented with 3 or, as in the following example, with 5 digits.

Example of the MODBUS function code

In the case of SCADA Software Lookout from National Instruments the MODBUS function codes are used with a 6 bit coding, whereby the first bit represents the function code:

Input code:	MODBUS function code	
0	FC1 ⇔ read coils	Reading of several input bits
1	FC2 ⇔ read input discretes	Reading of several input bits
3	FC3 ⇔ read multiple registers	Reading of several input registers
4	FC4 ⇔ read input registers	Reading of an individual input register

The following five digits specify the channel number of the consecutively numbered digital or analog input and/or output channels.

Examples:

- Read the first digital input: i.e. 0 0000 1
- Read the second analog input: i.e. 3 0000 2

Application example:

Thus, the digital input channel 2 of the above node "Measuring data" can be read out with the input: "Measuring data. 0 0000 2".

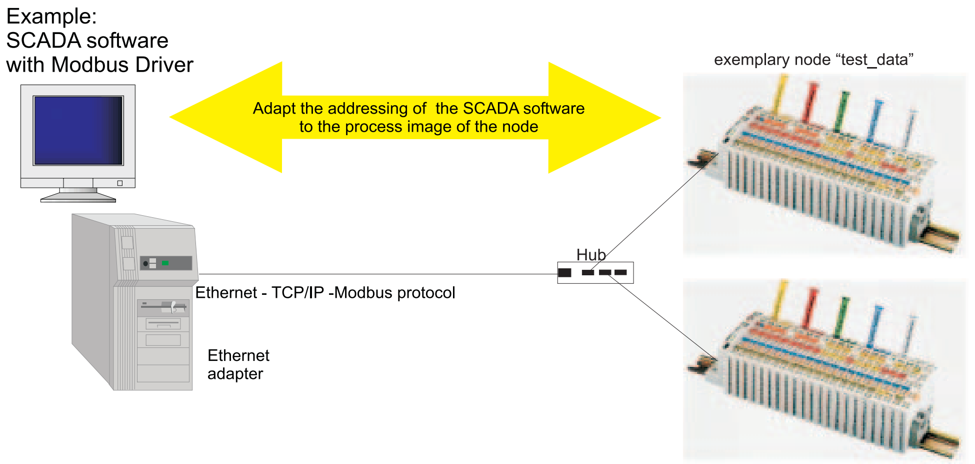


Fig. 6-1: Example of user software G012913e

More information
Please refer to the respective SCADA product manual for a detailed description of the particular software operation.

7 Use in Hazardous Environments

7.1 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.

7.2 Protective measures

Primarily, explosion protection describes how to prevent the formation of an explosive atmosphere. For instance by avoiding the use of combustible liquids, 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.

7.3 Classification meeting CENELEC and IEC

The specifications outlined here are valid for use in Europe and are based on the following standards: EN50... 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).

7.3.1 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).

7.3.2 Explosion protection group

In addition, the electrical components for explosive areas are subdivided into two groups:

Group I: Group I includes electrical components for use in fire-damp endangered mine structures.

Group II: Group II includes electrical components for use in all other explosive environments. This 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

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 group IIC is requested for maximum safety.

7.3.3 Unit categories

Moreover, the areas of use (zones) and the conditions of use (explosion groups) are subdivided into categories for the electrical operating means:

Unit categories	Explosion group	Area of use
M1	I	Fire-damp protection
M2	I	Fire-damp protection
1G	II	Zone 0 Explosive environment by gas, fumes or mist
2G	II	Zone 1 Explosive environment by gas, fumes or mist
3G	II	Zone 2 Explosive environment by gas, fumes or mist
1D	II	Zone 20 Explosive environment by dust
2D	II	Zone 21 Explosive environment by dust
3D	II	Zone 22 Explosive environment by dust

7.3.4 Temperature classes

The maximum surface temperature for electrical components of explosion protection 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 surface temperature for all ignition protection types, the electrical components are subdivided into temperature classes, as far as electrical components of explosion protection group II are concerned. Here the temperatures refer to a surrounding temperature of 40 °C for operation and testing of the electrical components. The lowest ignition temperature of the existing explosive atmosphere must be higher than the maximum surface temperature.

Temperature classes	Maximum surface temperature	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:

<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

7.3.5 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:

Identifi-cation	CENELEC standard	IEC standard	Explanation	Application
EEx o	EN 50 015	IEC 79-6	Oil encapsulation	Zone 1 + 2
EEx p	EN 50 016	IEC 79-2	Overpressure encapsulation	Zone 1 + 2
EEx q	EN 50 017	IEC 79-5	Sand encapsulation	Zone 1 + 2
EEx d	EN 50 018	IEC 79-1	Pressure resistant encapsulation	Zone 1 + 2
EEx e	EN 50 019	IEC 79-7	Increased safety	Zone 1 + 2
EEx m	EN 50 028	IEC 79-18	Cast encapsulation	Zone 1 + 2
EEx i	EN 50 020 (unit) EN 50 039 (system)	IEC 79-11	Intrinsic safety	Zone 0 + 1 + 2
EEx n	EN 50 021	IEC 79-15	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. The standard EN 50 021 allows electrical component manufacturers to obtain certificates from the corresponding authorities for instance KEMA in the Netherlands or the PTB in Germany, certifying that the tested components meet the above mentioned standards draft.

Type “n” ignition protection additionally requires electrical components to be marked with the following extended identification:

A – non spark generating (function modules without relay /without switches)

AC – spark generating, contacts protected by seals (function modules with relays / without switches)

L – limited energy (function modules with switch)

Further information
For more detailed information please refer to the national and/or international standards, directives and regulations!

7.4 Classifications meeting the NEC 500

The following classifications according to NEC 500 (National Electric Code) are valid for North America.

7.4.1 Divisions

The "Divisions" describe the degree of probability of whatever type of dangerous situation occurring. Here the following assignments apply:

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).

7.4.2 Explosion protection groups

Electrical components for explosion endangered areas are subdivided in three danger categories:

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

7.4.3 Temperature classes

Electrical components for explosive areas are differentiated by temperature classes:

Temperature classes	**Maximum surface temperature**	**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

7.5 Identification

7.5.1 For Europe

According to CENELEC and IEC

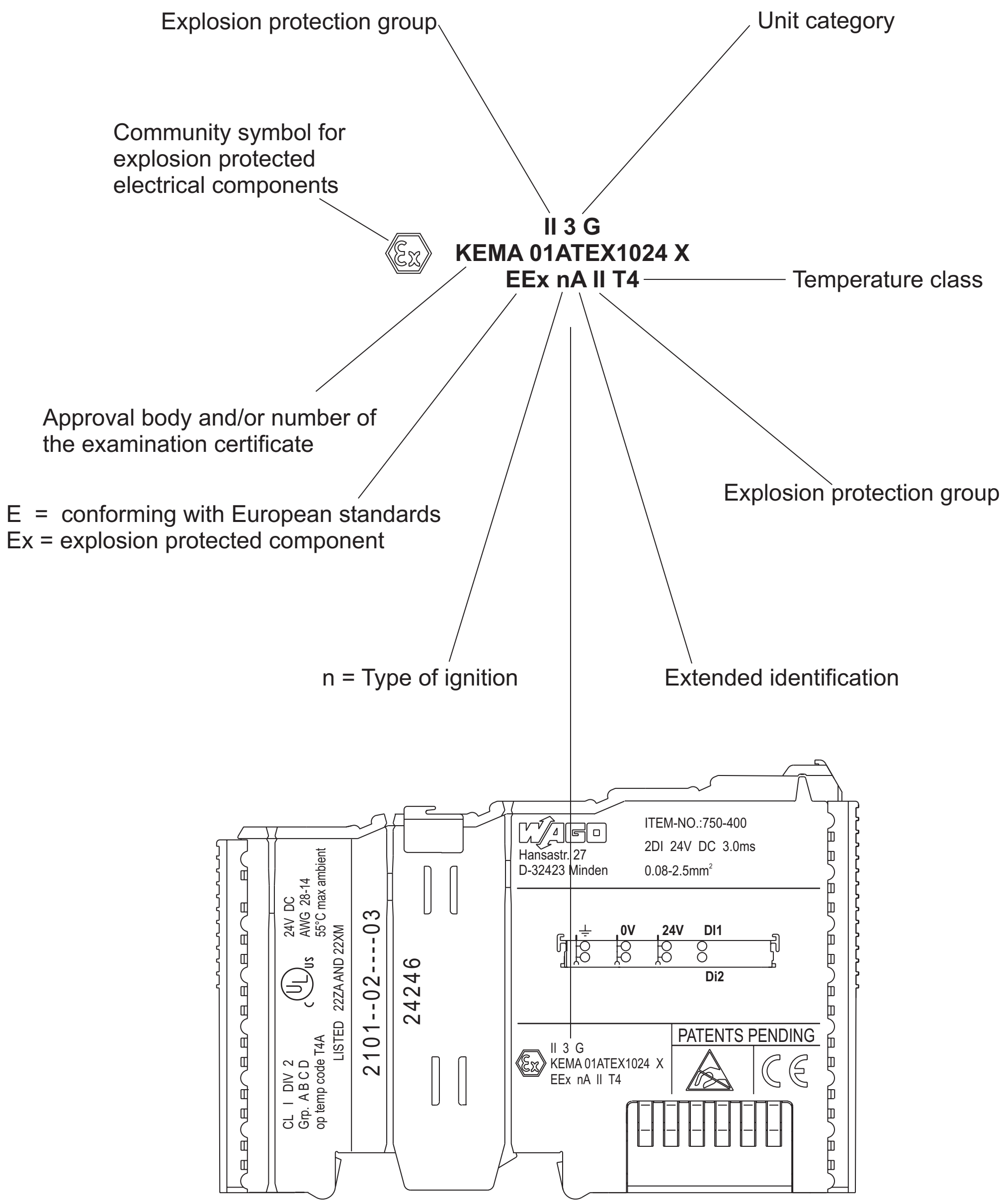


Fig. 7.5.1-1: Example for lateral labeling of bus modules
(750-400, 2 channel digital input module 24 V DC) g01xx03e

7.5.2 For America

According to NEC 500

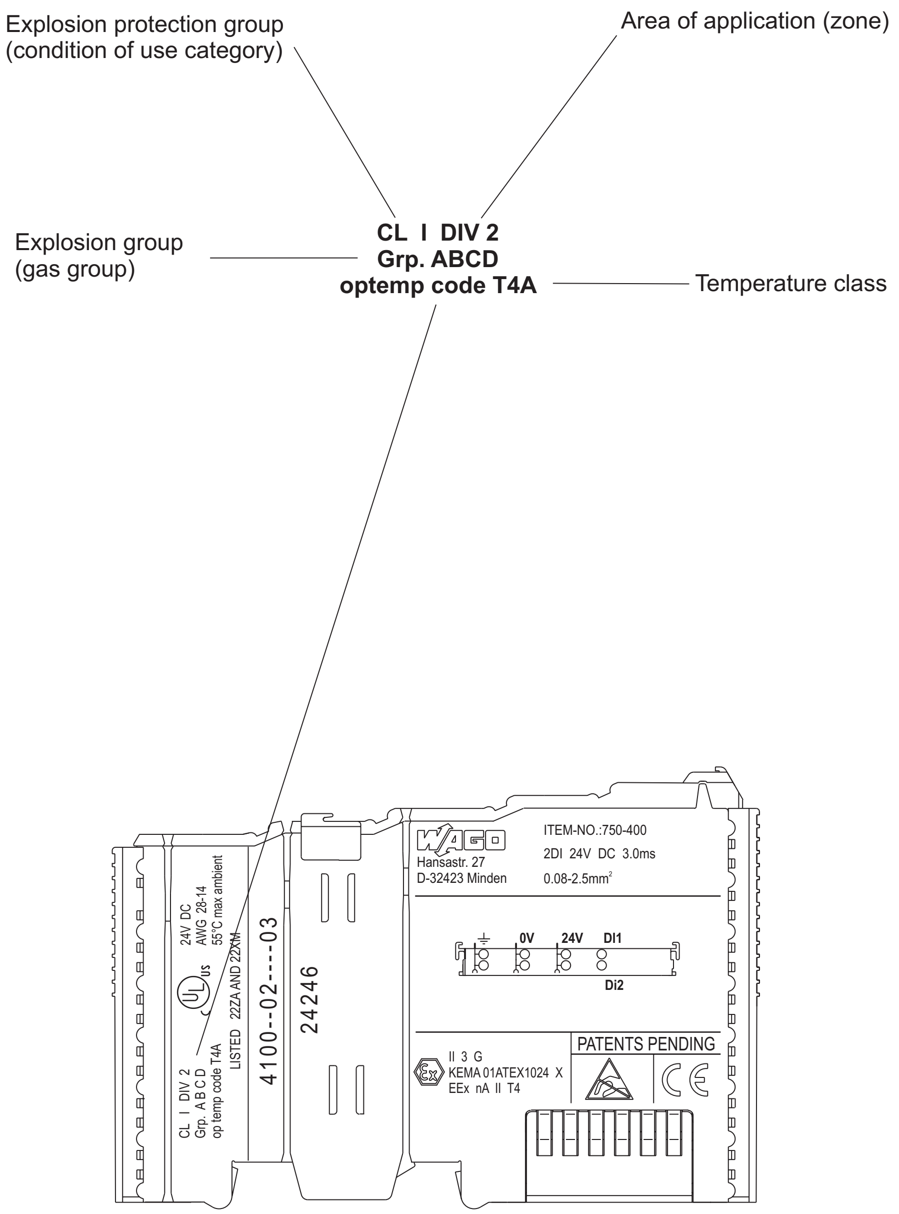


Fig. 7.5.2-1: Example for lateral labeling of bus modules
(750-400, 2 channel digital input module 24 V DC) g01xx04e

7.6 Installation regulations

In the **Federal Republic of Germany,** various national regulations for the installation in explosive areas must be taken into consideration. The basis being the ElexV complemented by the installation regulation DIN VDE 0165/2.91. The following are excerpts from additional VDE 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:

NFPA 70	National Electrical Code Art. 500 Hazardous Locations
ANSI/ISA-RP 12.6-1987	Recommended Practice
C22.1	Canadian Electrical Code

Danger
When using the WAGO-I/O SYSTEM 750 (electrical operation) with Ex approval, the following points are mandatory:

The fieldbus independent I/O System Modules Type 750-xxx are to be installed in enclosures that provide for the degree of ingress protection of at least IP54.
For use in the presence of combustible dust, the above mentioned modules are to be installed in enclosures that provide for the degree of ingress protection of at least IP64.

The fieldbus independent I/O system may only be installed in hazardous areas (Europe: Group II, Zone 2 or America: Class I, Division 2, Group A, B, C, D) or in non-hazardous areas!

Installation, connection, 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.

Ensure that only approved modules of the electrical operating type will be used. The Substitution or Replacement of modules can jeopardize the suitability of the system in hazardous environments!

Operation of intrinsically safe EEx i modules with direct connection to sensors/actuators in hazardous areas of Zone 0 + 1 and Division 1 type requires the use of a 24 V DC Power Supply EEx i module!

DIP switches and potentiometers are only to be adjusted when the area is know to be non-hazardous.

Further Information
Proof of certification is available on request. Also take note of the information given on the module technical information sheet.

8 Glossary

B	

Baseband

Systems which operate without carrier frequencies, i.e. with unmodulated signals. Therefore, they only offer one channel which has to be logically tailored to the various requirements. Opposite: Wideband.

Bit

Smallest information unit. Its value can either be 1 or 0.

Bit rate

Number of bits transmitted within a time unit.

BNC

Bayonet Navy Connector. Socket for coaxial cable.

BootP

the bootstrap protocol is a protocol which specifies how system and network information is to be transmitted from a *server* to work stations.

Bridge

Connects two separate networks.

Broadcast

A message that is sent to all station connected to the network.

Bus

A structure used to transmit data. There are two types, serial and parallel. A serial bus transmits data bit by bit, whereas a parallel bus transmits many bits at one time.

Byte

Binary Yoked Transfer Element. A byte generally contains 8 bits.

C	

Client

A system that requests the services of another. 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 a single wire and a radial shield to transmit information.

CSMA/CD

Carrier Sense Multiple Access with Collision Detection. When a collision is detected, all subscribers back off. After waiting a random delay time, the subscribers attempt to re-transmit the data.

D	

Data bus

see *Bus*.

Deterministic ETHERNET

The ETHERNET data is transferred at a defined time constant. The ETHERNET network can be defined and calculated. A *Switched ETHERNET* architecture makes this possible.

Driver

Software code which communicates with a hardware device. This communication is normally performed by internal device registers.

E	

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

In 1983 ETHERNET was standardized by *IEEE 802.3* 10Base-5. ISO took over the standardization in the ISO Standard 8802/3. The essential differences between ETHERNET and the IEEE standard are to be found in the frame architecture and treatment of pad characters.

F

Fieldbus

System for serial information transmission between devices of automation technology in the process-related field area.

Firewall

Collective term for solutions which protect *LANs* connection to the *Internet* from unauthorized access. 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 *subscriber*.

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 allows users on one machine to transfer files to/from another.

Function

Module that always returns the same result (as a function value), prerequisite being identical input values; it has no local variables that store values beyond an invoke.

Function block

Module that delivers one or more values when being executed. They can be stored as local variables („Memory“).

G

Gateway

Device for connecting two different networks. It converts the different protocols.

H

Hardware

Electronic, electrical and mechanic components of a module/subassembly.

Header

A portion of the data packet, containing, among others, the address information of the receiver.

Host computer / Subscriber

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, this term is also used to refer to simple computers which provide particular central *Services* (i.e. UNIX-Subscribers on the *Internet*).

HTML

Abbreviation of hypertext markup language
HTML is the description 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 exchanging HTML documents. It normally uses *port* 80.

Hub

A device which allows communication between several network users via *twisted pair* cable.
Similar to a *repeater*, but with many outputs, a hub is used to form a star topology.

Hypertext

Document format used by *HTTP*. Hypertext documents are text files which allow links to other text documents via particularly highlighted keywords.

I

IAONA Europe

IAONA Europe (Industrial Automation Open Networking Alliance) is an organization for industrial network technology with the objective to establish ETHERNET in automation technology.
Further information on this subject is available on the Internet under: www.iaona-eu.com.

ICMP-Protocol

TA protocol for the transmission of status information and error messages of the *IP*, *TCP* and *UDP* protocols between IP network nodes. ICMP offers, among others, the possibility of an echo (ping) request to determine whether a destination is available and is responding.

IEC 61131-3

International standard published in 1993 for morn systems with PLC functionality. Based on a structured software model, it defines a number of high performance programming languages that can be used for various automation tasks.

IEEE

Institute of Electrical and Electronic Engineers.

IEEE 802.3

IEEE 802.3 is a IEEE standard. ETHERNET only supports the yellow cable as a medium. IEEE 802.3 also supports *S-UTP* and wideband coaxial cable. The segment lengths range from 500 m for yellow cable, 100 m for TP and 1800 m for wideband coaxial cable. A star or a bus topology is possible. ETHERNET (IEEE 802.3) uses *CSMA/CD* as a channel access method.

Intel format

Set configuration of the fieldbus coupler / controller to establish 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*,...). The format determines whether or not high and low bytes are changed over. They are not changed over with the Intel format.

Internet

A collection of networks interconnected to each other throughout the world. Its most well known area is the *World Wide Web*.

Intranet

A network concept with private network connections over which data can be exchanged within a company.

IP

Internet Protocol. The connectionless network layer, which relies on upper protocols to provide reliability.

ISA

Industry Standard Architecture. Offers a standard interface for the data exchange between CPU and periphery.

ISO/OSI- Reference Model

Reference model of the ISO/OSI for networks with the objective 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.

L

LAN

Local Area Network

Library

Compilation of modules available to the programmer in the programming tool **WAGO-I/O-*PRO 32*** for the creation of a control program according to IEC 61131-3.

M

Mail Server

Internet E-mails are transported and stored temporarily by so-called Mail servers. The personal post can be downloaded by such a Mail server or be sent in reverse to the far dispatch to these. With the SMTP protocol E-mails can be dispatched.

Manchester encoding

In this encoding system, a 1 is encoded as a transition from *low* to *high* and a 0 as a transition from *high* to *low*.

Modules

Functions, function blocks and programs are modules.
Each module has a declaration part and a body, the latter being written in one of the IEC programming languages IL (instruction list), ST (structured text), SFC (sequential flow), FBD (function block diagram) or LD (ladder diagram).

MS-DOS

Operating system, which allows all applications direct access to the hardware.

O

Open MODBUS/TCP Specification

Specification which establishes the specific structure of a MODBUS/TCP data packet. This is dependant upon the selected function code.

Operating system

Software which links the application programs to the hardware.

P

Ping command

When a ping command (ping <IP address>) is entered, the ping program *ICMP* generates echo *request* packets. It is used to test whether a node is available.

Port number

The port number, together with the IP address, forms an unambiguous connection point between two processes (applications).

Predictable ETHERNET

The delay time of a message on an ETHERNET network can be predicted. The measures which have been taken in predictable ETHERNET make it virtually possible to realize realtime requirements.

Proxy gateway

A proxy gateway (or proxy *server*, too) allows systems which do not have direct access to the *Internet*, indirect access to the network. These can be systems which are excluded from direct access by a *firewall* for security reasons.
A proxy can filter out individual data packets between the Internet and a local network to increase security. Proxies are also used to limit access to particular servers.
In addition, proxy gateways can also have a cache function, in which case they check whether the respective *URL* address is already available locally and return it immediately, if necessary. This saves time and costs when there are multiple accesses. If the URL is not in the cache, the proxy forwards the *request* as normal.
The user should not notice the proxy *gateway* 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 are physical amplifiers without their own processing function. They refresh data without detecting damaged data and forward all signals. Repeaters are used for longer transmission distances or when the maximum number of nodes of 64 devices per *twisted pair* segment is exceeded. A request from a client to server is a provision to act on a serivce or funtion call.

Request

A service request from a client which requests the provision of a service from a server.

Response

The server’s reply to a client’s request.

RFC specifications

Specifications, suggestions, ideas and guidelines regarding the *Internet* are published in the form of RFCs (Request For Comments).

RJ45 connector

Also referred to as a Western connector. This connector allows the connection of two network controllers via *twisted pair* cables.

Router

Connects neighboring *subnets*, 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 as *repeaters* or *bridges*.

Routing

Method of selecting the best path over which to send data to a distant network.

S

SCADA

Abbreviation for Supervisory Control and Data Acquisition. SCADA software is a program for the control and visualization of processes.

Segment

Typically, a network is divided up into different physical network segments by way of *routers* or *repeaters*.

Server

Device providing services within a client/server system. The service is requested by the *Client*.

Service

An operation targeted at an object (read, write).

SMTP

Short form for „Simple Mail Transfer Protocol“. Standard protocol, with which E-mails are sent away in the internet.

SOAP

Short form for “Simple Object Access Protocol“. XML is a standard for Meta data, the access on the XML objects takes place via SOAP.The standard defines, how transactions via internet and XML can be done and how dynamic Web services over distributed networks can be used.

Socket

Is a software interface introduced with BSD-UNIX for inter-process communication. Sockets are also possible in the network via TCP/IP. As from Windows 3.11, they are also available in Microsoft operating systems.

STP

With the STP cable (Shielded twisted pair) it acts around a symmetrical cable with in pairs stranded and protected veins. The classical 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 screen.

S-STP

Beside the STP cables there is cable, which has total shielding from foil or network shielding additionally to the single shielding of the conductors still another. These cables are called S/STP cables: Screened/Shielded twisted pair.

Structured cabling

This specifies the maximum permissible cable lengths (EIA/TIA 568, IS 11801) and gives recommendations for the different types topology for ground area, building and floor cabling.

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

The subnet mask can be used to manipulate the address areas in the IP address room with reference to the number of *subnets* and *subscribers*. A standard subnet mask is, for example, 255.255.255.0.

S-UTP

Screened unshielded *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. A switch switches a virtual connection between an input port and an output port for data transmission. Switches learn which nodes are connected and filter the information transmitted over the network accordingly. Switches are inteligent devices that learn the node connections and can transfer data at the switch and not have to send it back to the main server.

Switched ETHERNET

The segments of this type of ETHERNET are connected by *switches*. There are many applications for switching technologies. ETHERNET switching is becoming increasingly popular in local networks as it allows the realization of a *deterministic ETHERNET*.

T

TCP

Transport Control Protocol.

TCP/IP Protocol Stack

Network protocols which allow communication between different networks and technologies.

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.

Twisted Pair

Twisted pair cables (abbreviated to TP).

U

UDP protocol

The user datagram protocol is a transport protocol (layer 4) of the *ISO/OSI-reference model* which supports data exchange between computers without a connection. UDP runs directly on top of the underlying *IP* protocol.

URL

Abbreviation for uniform resource locator.
Address form for *Internet* files which are mostly applied within the World Wide Web (*WWW*). The URL format makes the unambiguous designation of all documents on the Internet possible by describing the address of a document or object which can be read by a *web browser*. URL includes the transmission type (http, ftp, news etc.), the computer which contains the information and the path on the computer. URL has the following format: Document type//Computer name/List of contents/File name.

UTP

The UTP cable is a symmetrical, not-protected cable with twisted colored wires in pairs. This type of cable, which there is in execution two-in pairs and four-in pairs, is the dominating type of cable in the floor wiring and the terminal wiring.

W

WAGO-I/O-*PRO* CAA

Uniform programming environment, programming tool from WAGO Kontakttechnik GmbH for the creation of a control program according to IEC 61131-3 for all programmable fieldbus controllers. Allows testing, debugging and the start-up of a program.

Web browser

Program for reading *hypertext*. The browser allows the various documents to be viewed in hypertext and navigation between documents.

Wide band

Transmission technology which operates with a high bandwidth, thereby permitting high transmission rates. This allows several devices to transmit simultaneously.
Opposite: Baseband.

Word-alignment

Set configuration of the fieldbus coupler/controller for the creation of a process image. Word-alignment is used to establish the process image word-by-word (2 bytes).

World Wide Web

HTTP server on the Internet.

9 Literature List

Manual TCP/IP-ETHERNET for Beginners
Wiesemann & Theis GmbH
1st edition, 11/1999
http://www.WuT.de/us_printmed.html

Local Area Networks - An introduction to the technology
John E. McNamara,
Digital Press, 1985
ISBN 0-932376-79-7 Digital Press Teil Nummer EY-00051-DP

Network Troubleshooting Guide von Digital Equipment Corporation,
August 1990,
Digital Press Teil Nummer EK-339AB-GD-002

Zu RFC:
Request for Comments
http://members.xoom.com/spielchen2k/archiv/public/exploits/rfcs/rfcs/

10 Index

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