Manual

WAGO-I/O-SYSTEM 750

Programmable Fieldbus Controller ETHERNET

750-881

10/100 Mbit/s; digital and analog Signals

Version 1.1.3

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

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

E-Mail: info@wago.com

Web: http://www.wago.com

Technical Support

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

E-Mail: support@wago.com

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

E-Mail: documentation@wago.com

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

Table of Contents

1 Notes about this Documentation

Note

Keep this documentation!
The operating instructions are part of the product and shall be kept for the entire lifetime of the device. They shall be transferred to each subsequent owner or user of the device. Care must also be taken to ensure that any supplement to these instructions are included, if applicable.

1.1 Validity of this Documentation

This documentation is only applicable to the device:
"Programmable Fieldbus Controller ETHERNET" 750-881 of the
WAGO-I/O-SYSTEM 750 series.

The Programmable Fieldbus Controller ETHERNET 750-881 shall only be installed and operated according to the instructions in this manual and the system description for the WAGO-I/O-SYSTEM 750.

NOTICE

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

1.2 Copyright

1.3 Symbols

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


Personal Injury Caused by Electric Current!
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

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

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

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


Damage to Property Caused by Electrostatic Discharge (ESD)!
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.


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

Information

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

1.4 Number Notation

Table 1: Number Notation

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

1.5 Font Conventions

Table 2: Font Conventions

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

2 Important Notes

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

2.1 Legal Bases

2.1.1 Subject to Changes

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

2.1.2 Personnel Qualifications

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

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

2.1.3 Use of the 750 Series in Compliance with Underlying Provisions

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

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

Operating 750 Series components in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section on "WAGO-I/O-SYSTEM 750" → "System Description" → "Technical Data" in the manual for the used fieldbus coupler/controller.

Appropriate housing (per 94/9/EG) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

2.1.4 Technical Condition of Specified Devices

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

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

2.2 Safety Advice (Precautions)

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

DANGER

Do not work on components while energized!
All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.

DANGER

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

NOTICE

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

NOTICE

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

NOTICE

Cleaning only with permitted materials!
Clean soiled contacts using oil-free compressed air or with ethyl alcohol and leather cloths.

NOTICE

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

NOTICE

Do not reverse the polarity of connection lines!
Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

NOTICE

Avoid electrostatic discharge!
The devices are equipped with electronic components that you may destroy by electrostatic discharge when you touch. Pay attention while handling the devices to good grounding of the environment (persons, job and packing).

2.3 Special Use Conditions for ETHERNET Devices

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

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

3 System Description

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

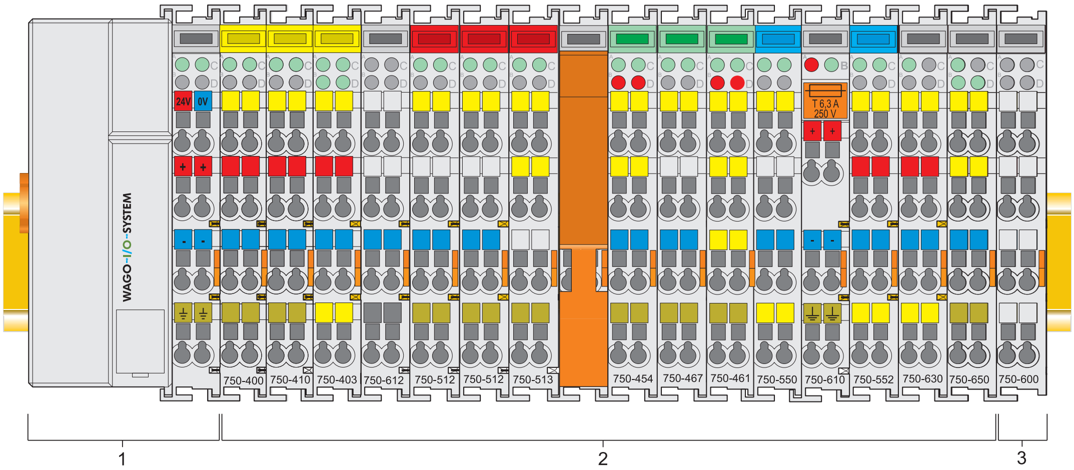


Figure 1: Fieldbus node

Couplers/controllers are available for different fieldbus systems.

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

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

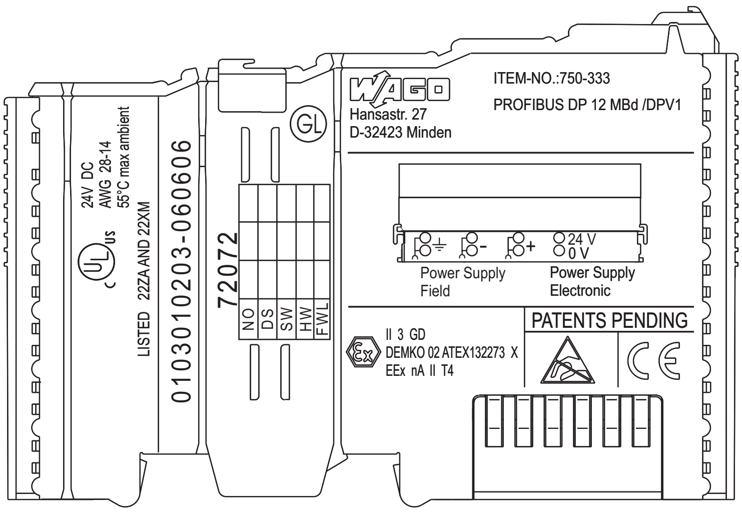
Bus modules for diverse digital and analog I/O signals as well as special functions can be connected to the fieldbus coupler/controller. The communication between the fieldbus coupler/controller and the bus modules is carried out via an internal bus.

The WAGO-I/O-SYSTEM 750 has a clear port level with LEDs for status indication, insertable mini WSB markers and pullout group marker carriers for marking.

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

3.1 Manufacturing Number

The serial number indicates the delivery status directly after production. This number is part of the labeling on the side of each component.
In addition, the serial number is printed on the cover cap of the configuration and programming interface of the fieldbus coupler/controller, so that it can also be read when installed.



Manufacturing number

01	**03**	**01**	**02**	**03**	**-B060606**
Calendar week	Year	Software version	Hardware version	Firmware loader version	Internal number

Figure 2: Example of a manufacturing number

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

3.2 Hardware Address (MAC ID)

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

3.3 Component Update

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

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

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

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

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

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

3.4 Storage, Assembly and Transport

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

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

3.5 Assembly Guidelines/Standards

DIN 60204	Electrical equipping of machines
DIN EN 50178	Equipping of high-voltage systems with electronic components (replacement for VDE 0160)
EN 60439	Low voltage switchgear assemblies

3.6 Power Supply

3.6.1 Isolation

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

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

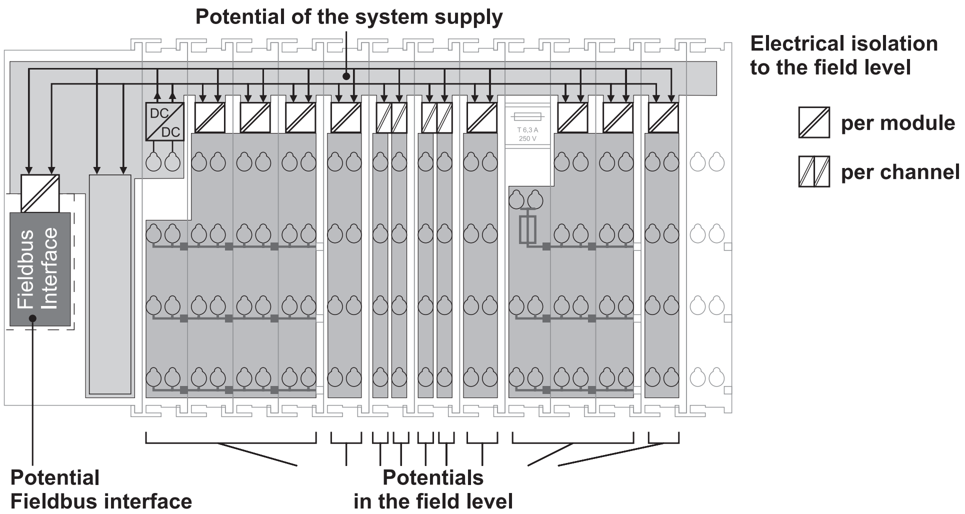


Figure 3: Isolation for Standard Couplers/Controllers and extended ECO Couplers

Note

Ensure protective conductor function is present (via ring feeding if required)!

Pay attention that the ground wire connection must be present in each group. In order that all protective conductor functions are maintained under all circumstances, it is recommended that a ground wire be connected at the beginning and the end of a potential group (please see chapter “Grounding” > “Grounding Protection”, Ring Feeding). Thus, if a bus module comes loose from a composite during servicing, therefore the protective conductor connection is still guaranteed for all connected field devices.

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

3.6.2 System Supply

3.6.2.1 Connection

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

NOTICE

Do not use an incorrect voltage/frequency!
The use of an incorrect supply voltage or frequency can cause severe damage to the component.

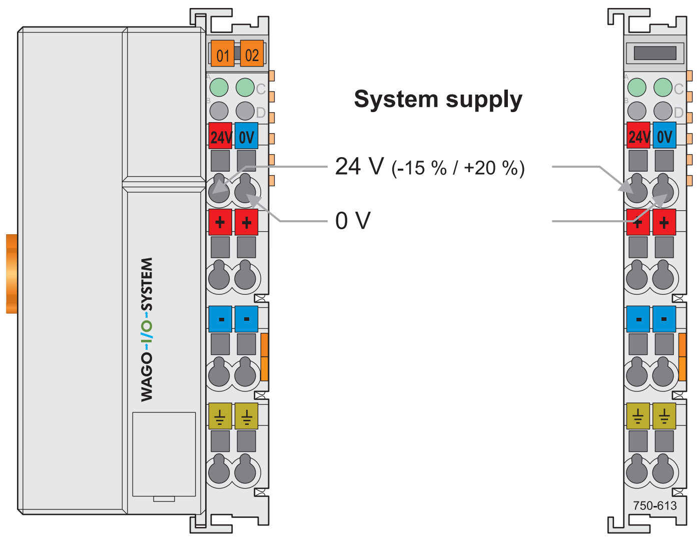


Figure 4: System supply for standard coupler/controller and extended ECO couplers

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

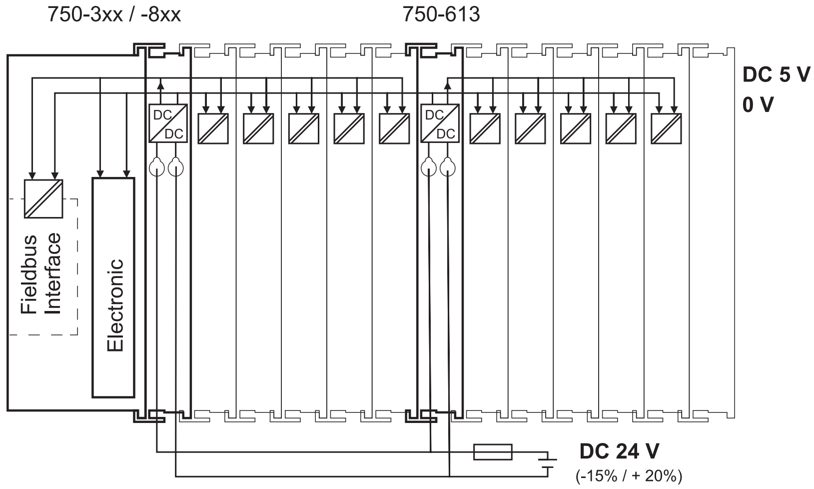


Figure 5: System voltage for standard couplers/controllers and extended ECO couplers

Note

Only reset the system simultaneously for all supply modules!
Reset the system by simultaneously switching the system supply on all supply modules (fieldbus coupler/controller and potential supply module with bus power supply 750-613) off and on again.

3.6.2.2 Dimensioning

Note

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

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

Table 3: Alignment

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

*) See current catalog, manuals, Internet

Example:

Calculating the current consumption on an Example Coupler:

Internal current consumption	380 mA at 5 V
Residual current for bus modules	1620 mA at 5 V
Sum $I_{(5\ V)\ total}$	**2000 mA at 5V**

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

Note

Observe total current of I/O modules, re-feed the potential if required!
If the sum of the internal current consumption exceeds the residual current for bus modules, then an internal system supply module (750-613) must be placed before the module where the permissible residual current was exceeded.

Example:

Calculating the total current on the Example Coupler described above:

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

Internal current consumption	20 * 90 mA =	1800 mA
	10 * 2 mA =	20 mA
Sum		**1820 mA**

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

Note

Recommendation
You can configure with the WAGO ProServe® Software **smartDESIGNER**, the assembly of a fieldbus node. You can test the configuration via the integrated plausibility check.

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

Coupler or controller

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

Internal system supply module 750-613

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

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

$\eta = 0.87$
(87 % Efficiency of the power supply at nominal load 24 V)

Note

Activate all outputs when testing the current consumption!
If the electrical consumption of the power supply point for the 24 V system supply exceeds 500 mA, then the cause may be an improperly designed node or a defect.
During the test, you must activate all outputs, in particular those of the relay modules.

3.6.3 Field Supply

3.6.3.1 Connection

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

The coupler/controller provides field side power (DC 24 V). In this case it is a passive power supply without protection equipment. Power supply modules are available for other potentials, e. g. AC 230 V.

Likewise, with the aid of the power supply modules, various potentials can be set up. The connections are linked in pairs with a power contact.

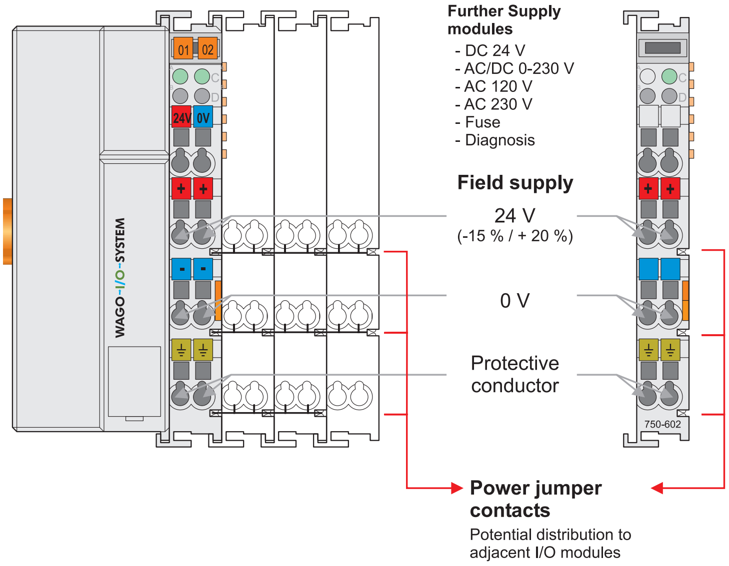


Figure 6: Field supply (sensor/actuator) for standard couplers/controllers and extended ECO couplers

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

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

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

Note

Re-establish the ground connection when the connection to the power jumper contacts is disrupted!
Some bus modules have no or very few power contacts (depending on the I/O function). Due to this, the passing through of the relevant potential is disrupted. If you require a field supply for subsequent bus modules, then you must use a power supply module.
Note the data sheets of the bus modules.

Note

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

3.6.3.2 Fusing

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

Table 4: Power supply modules

Order No.	Field Voltage
750-601	24 V DC, Supply/Fuse
750-609	230 V AC, Supply/Fuse
750-615	120 V AC, Supply/Fuse
750-610	24 V DC, Supply/Fuse/Diagnosis
750-611	230 V AC, Supply/Fuse/Diagnosis

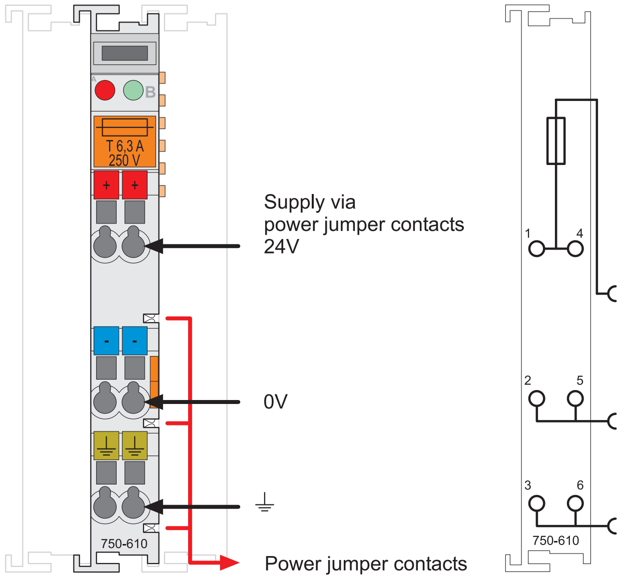


Figure 7: Supply module with fuse carrier (Example 750-610)

NOTICE

Observe the maximum power dissipation and, if required, UL requirements!
In the case of power supply modules with fuse holders, you must only use fuses with a maximum dissipation of 1.6 W (IEC 127).
For UL approved systems only use UL approved fuses.

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

Figure 8: Removing the fuse carrier

Lifting the cover to the side opens the fuse carrier.

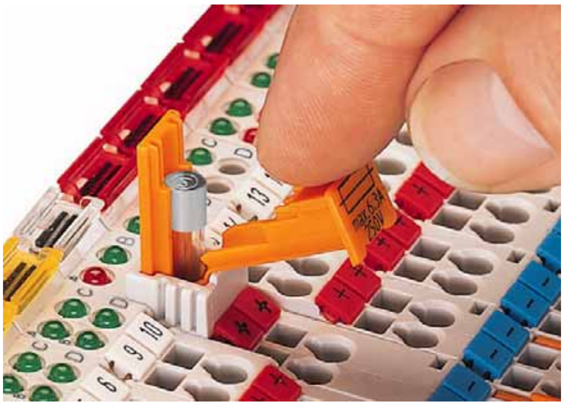


Figure 9: Opening the fuse carrier

Figure 10: Change fuse

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

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

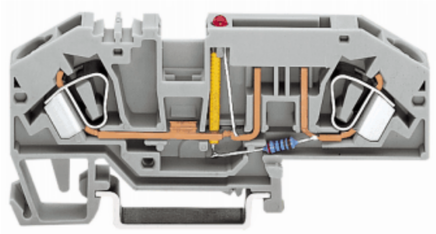


Figure 11: Fuse modules for automotive fuses, series 282

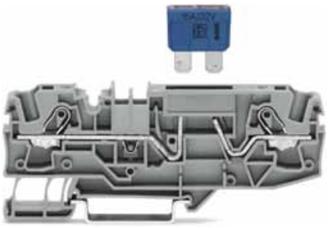


Figure 12: Fuse modules for automotive fuses, series 2006

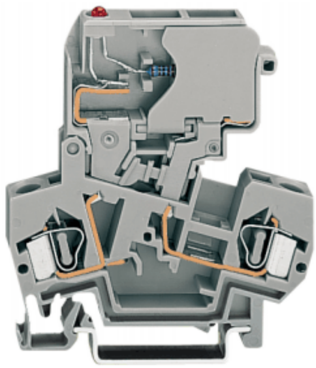


Figure 13: Fuse modules with pivotable fuse carrier, series 281

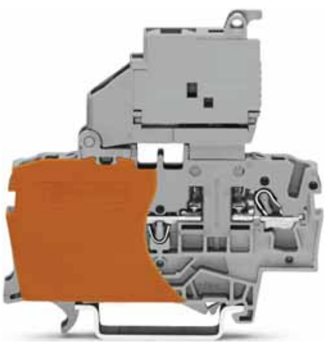


Figure 14: Fuse modules with pivotable fuse carrier, series 2002

3.6.4 Supplementary Power Supply Regulations

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

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

Table 5: Filter modules for 24-volt supply

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

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

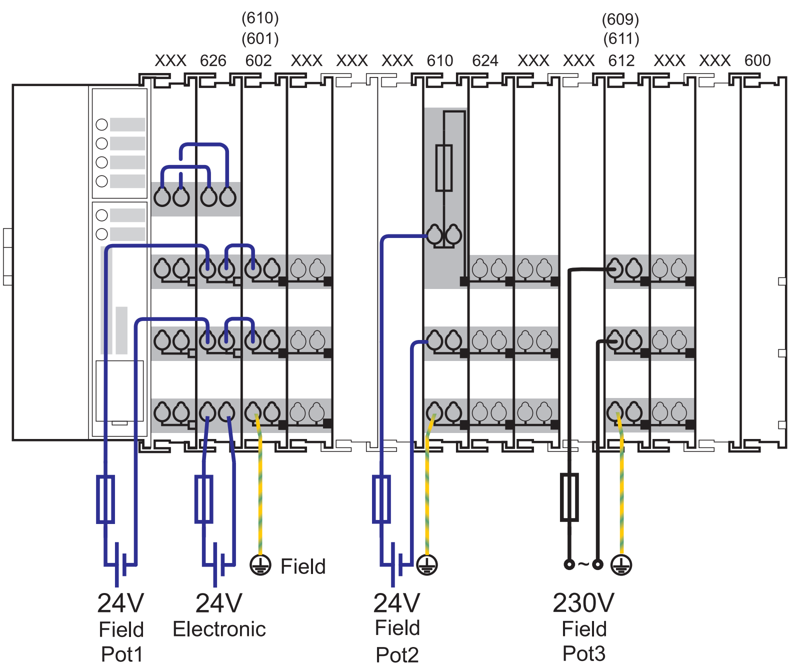


Figure 15: Power supply concept

Note

Additional supply module as ground (earth) conductor/fuse protection!
Only insert another potential power terminal 750-601/602/610 behind the filter terminal 750-626 if you need the protective earth conductor on the lower power contact or if you require a fuse protection.

3.6.5 Supply Example

Note

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

Information

Additional information about the ring feeding
In order to increase the system security, a ring feeding of the earth potential is recommended. Thus, if a bus module comes loose from a composite during servicing, then the protective conductor connection is still guaranteed for all connected field devices. With the ring feeding, protective grounding is connected at the beginning and the end of a potential group.
Please refer for further information to chapter „Grounding “> “Grounding Protection”, Ring Feeding.

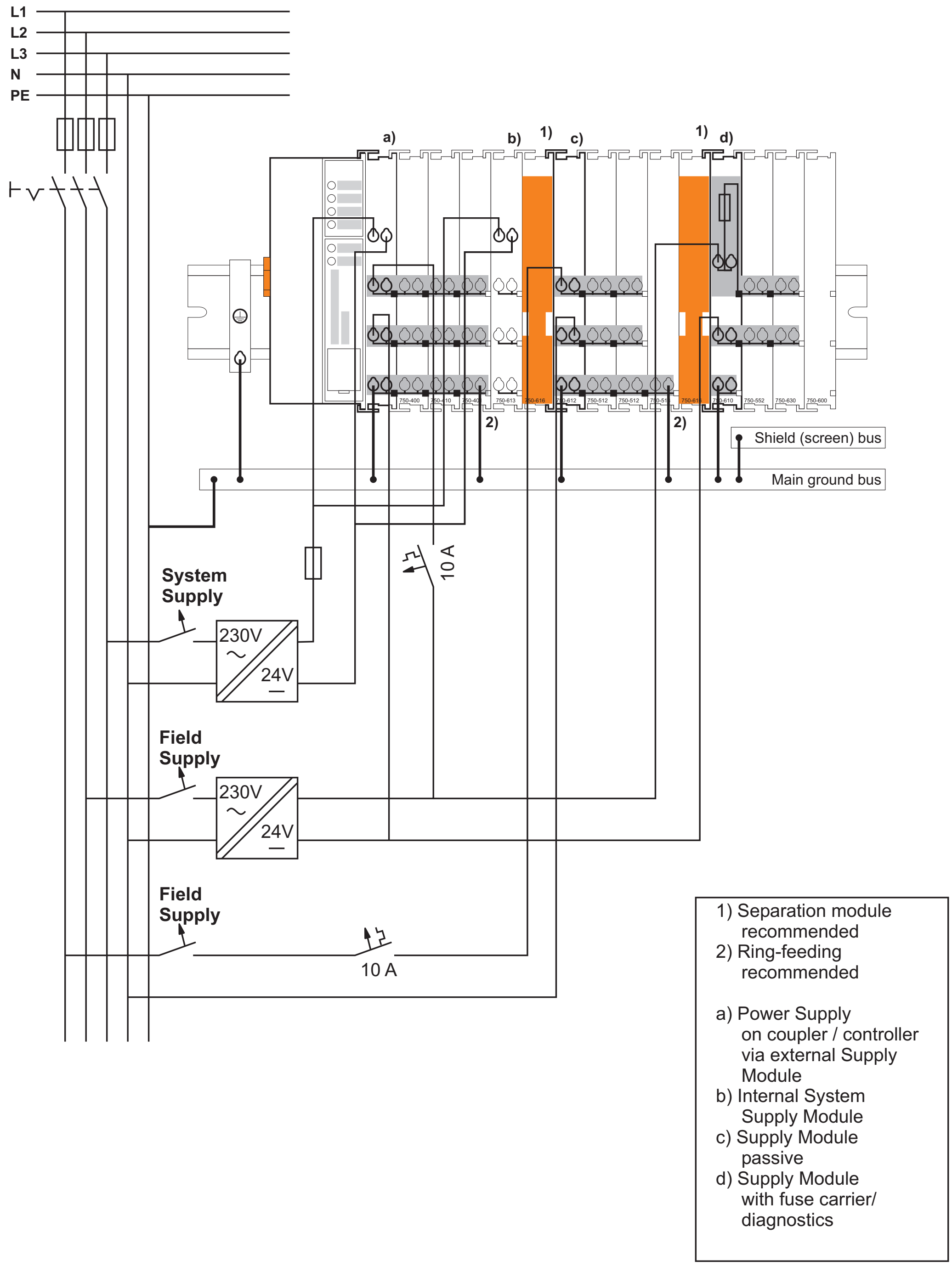


Figure 16: Supply example for standard couplers/controllers and extended ECO couplers

3.6.6 Power Supply Unit

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

Note

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

A buffer (200 µF per 1 A current load) should be provided for brief voltage dips.

Note

Power failure time is not acc. to IEC 61131-2!
Note that the power failure time in a node with maximal components is not 10 ms, according to the defaults of the IEC61131-2 standard.

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

Note

System and field supply shall be isolated from the power supply!
You should isolate the system supply and the field supply from the power supplies in order to ensure bus operation in the event of short circuits on the actuator side.

Table 6: WAGO Power Supply Unit

WAGO Power Supply Unit	Description
787-612	Primary switched mode; DC 24 V; 2,5 A Input nominal voltage AC 230 V
787-622	Primary switched mode; DC 24 V; 5 A Input nominal voltage AC 230 V
787-632	Primary switched mode; DC 24 V; 10 A Input nominal voltage AC 230/115 V
 288-809 288-810 288-812 288-813	Rail-mounted modules with universal mounting carrier AC 115 V/DC 24 V; 0,5 A AC 230 V/DC 24 V; 0,5 A AC 230 V/DC 24 V; 2 A AC 115 V/DC 24 V; 2 A

3.7 Grounding

3.7.1 Grounding the DIN Rail

3.7.1.1 Framework Assembly

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

DANGER

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

3.7.1.2 Insulated Assembly

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

Note

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

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

Table 7: WAGO ground wire terminals

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

3.7.2 Grounding Function

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

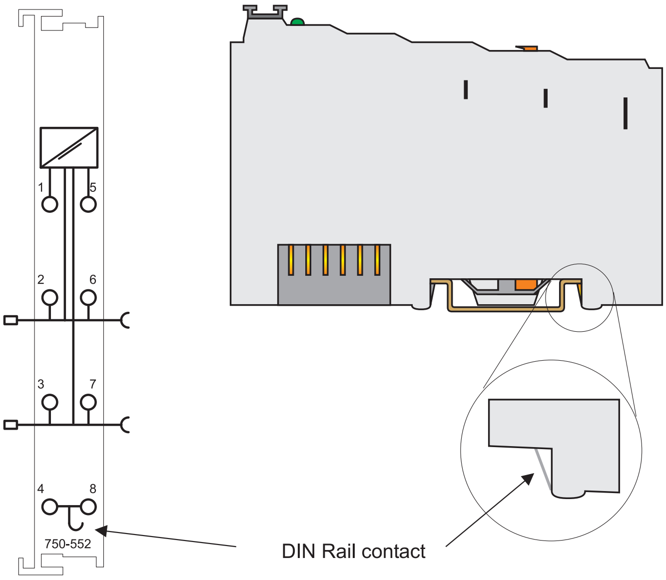


Figure 17: Carrier rail contact

DANGER

Ensure sufficient grounding is provided!
You must take care to ensure the direct electrical connection between the carrier rail contact and the carrier rail.
The carrier rail must be grounded.
For information on carrier rail properties, see chapter "Assembly onto Carrier Rail > Carrier Rail Properties".

3.7.3 Grounding Protection

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

Note

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

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

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

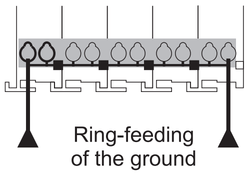


Figure 18: Ring-feeding

Note

Observe grounding protection regulations!
You must observe the regulations relating to the place of assembly as well as the national regulations for maintenance and inspection of the grounding protection.

3.8 Shielding

3.8.1 General

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

Note

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

Note

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

Note

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

3.8.2 Bus cables

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

3.8.3 Signal lines

I/O modules for analog signals and some interface I/O modules are equipped with shield clamps.

Note

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

3.8.4 WAGO Shield Connecting System

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

Figure 19: Example of the WAGO shield connecting system

Figure 20: Application of the WAGO shield connecting system

4 Device Description

The 750-881 programmable Fieldbus Controller (PFC) combines the functionality of an ETHERNET-based Fieldbus Coupler with the functionality of a Programmable Logic Controller (PLC).

This controller can be used for applications in machine and plant construction as well as in the process industry and building technology.

The two Ethernet interfaces and the integrated switch make possible the wiring of the fieldbus in line topology. Thus additional infrastructure elements such as switches or hubs can be void. Both interfaces support Autonegotiation and Auto-MDI (X).

With the DIP switch the last byte of the IP address, as well as the assignment of the IP address (DHCP, BootP, firm setting) can be given.

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

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

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

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

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

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

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

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

The fieldbus controller has 1 MB program memory, 512 KB data memory and 32 KB retentive memory available for the IEC 61131-3 programming.

The user can access all fieldbus and I/O data.

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

The MODBUS/TCP(UDP) protocol and the ETHERNET/IP protocol are implemented for exchanging process data. Both of these communication protocols can be used either together or separately. For this, the write access to the I/O modules (access via PFC, MODBUS/TCP or EtherNet/IP) is specified in an xml file.

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

For the data transfer via ETHERNET the FTP is available.

For the automatic assignment of the IP address in the network, kann alternatively DHCP or BootP can be used.

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

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

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

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

By default, the controller’s built-in HTML pages contain information on the configuration and status of the PFC, and can be read using a normal web browser. In addition, a file system is implemented that allows you to store custom HTML pages in the controller using FTP download or to store your own HTML pages or call up programs directly.

Table 8: Compatibility

Programming tool:	CoDeSys						
-Version	V2.3.9.19						
Fieldbus controller:							
750-881	✓						

Commentary:

✓	Fieldbus controller compatible with WAGO-I/O-*PRO* version, independent of the controller hard- or software.

4.1 View

The view below shows the three parts of the device:

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

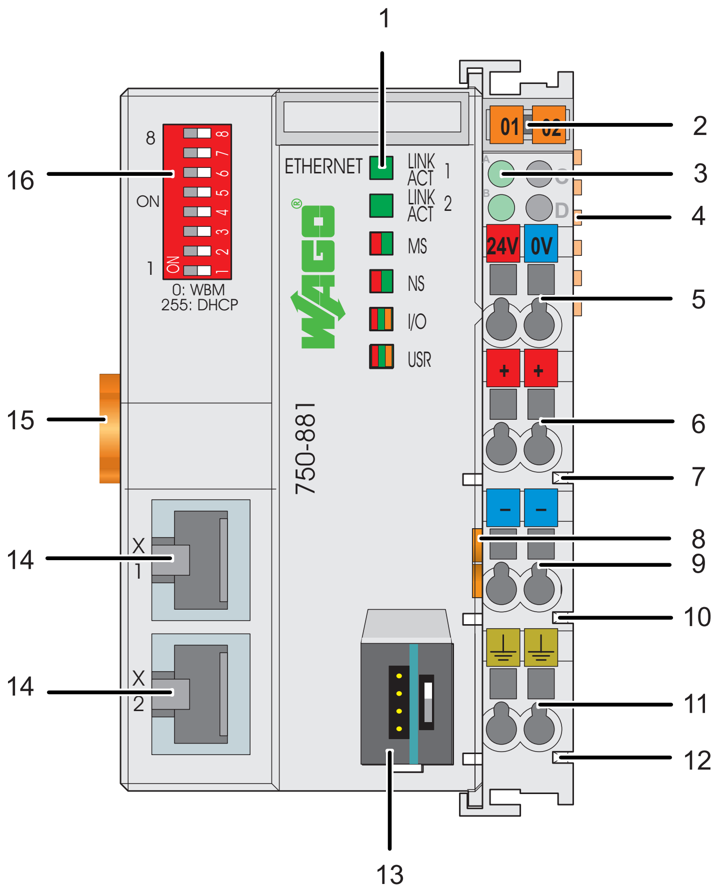


Figure 21: View ETHERNET TCP/IP Fieldbus Controller

Table 9: Legend to the View ETHERNET TCP/IP Fieldbus Controller

No.	Designation	Meaning	Details see Chapter:
1	LINK ACT 1, 2, MS, NS, I/O, USR	Status LEDs Fieldbus	"Device Description" > "Display Elements"
2	---	Group marking carrier (retractable) with additional marking possibility on two miniature WSB markers	---
3	A, B or C	Status LED's System/Field Supply	"Device Description" > "Display Elements"
4	---	Data Contacts	"Connect Devices" > "Data Contacts/Internal Bus"
5	24 V, 0 V	CAGE CLAMP® Connections System Supply	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
6	+	CAGE CLAMP® Connections Field Supply DC 24 V	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
7	---	Power Jumper Contact 24 V DC	"Connect Devices" > "Power Contacts/ Field Supply"
8	---	Unlocking Lug	"Assembly" > "Inserting and Removing Devices"
9	-	CAGE CLAMP® Connections Field Supply 0 V	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
10	---	Power Jumper Contact 0 V	"Connect Devices" > "Power Contacts/ Field Supply"
11	(Earth)	CAGE CLAMP® Connections Field Supply (Earth)	"Connect Devices" > "Connecting a conductor to the CAGE CLAMP®"
12	---	Power Jumper Contact (Earth)	"Connect Devices" > "Power Contacts/ Field Supply"
13	---	Service Interface (open flap)	"Device Description" > "Operating Elements"
14	X1, X2	Fieldbus connection 2 x RJ-45 as 2-Port ETHERNET Switch	"Device Description" > "Connectors"
15	---	Locking Disc	"Assembly" > "Inserting and Removing Devices"
16	---	Address Selection Switch	"Device Description" > "Operating Elements"

4.2 Connectors

4.2.1 Device Supply

The device is powered via terminal blocks with CAGE CLAMP® connections. The device supply generates the necessary voltage to power the electronics of the device and the internal electronics of the connected I/O modules.

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

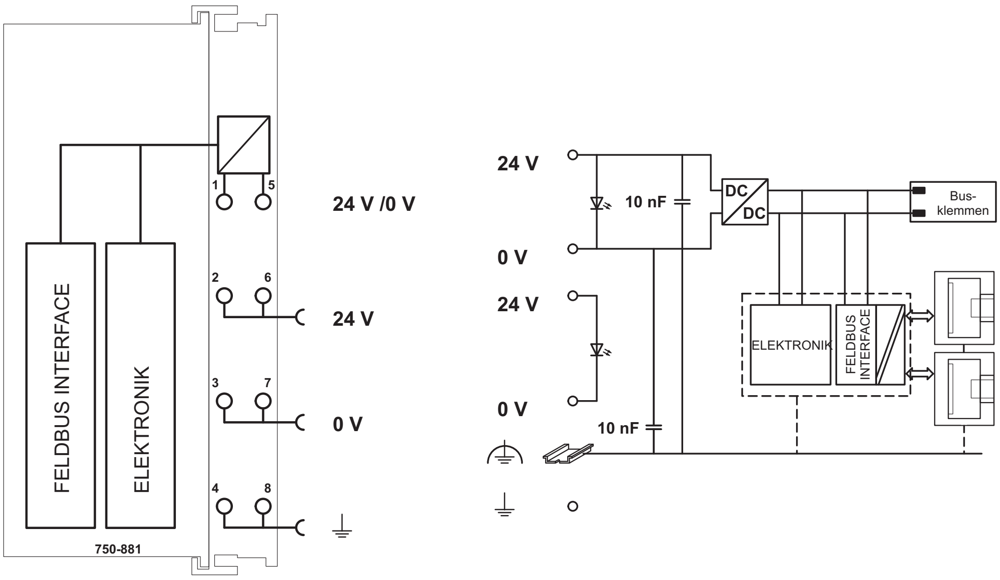


Figure 22: Device Supply

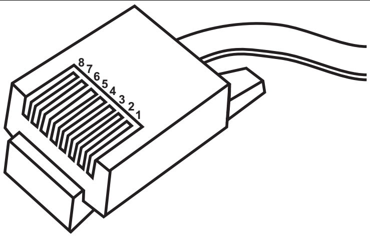
4.2.2 Fieldbus Connection

The connection to the fieldbus is made via two RJ-45 plugs (also called "Western plugs"), which are connected to the fieldbus controller via an integrated switch.
The integrated switch works in store-and-forward operation and for each port, supports the transmission speeds 10/100 Mbit as well as the transmission modes full and half-duplex.

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

The RJ-45 socket is physically lower, allowing the coupler to fit in an 80 mm high enclosure once connected.

Table 10: RJ-45 Connector and RJ-45 Connector Configuration

View	Contact	Signal	
 Figure 23: RJ-45-Connector	1	TD +	Transmit +
	2	TD -	Transmit -
	3	RD +	Receive +
	4		free
	5		free
	6	RD -	Receive -
	7		free
	8		free

NOTICE

Not for use in telecommunication circuits!
Only use devices equipped with ETHERNET or RJ-45 connectors in LANs.
Never connect these devices with telecommunication networks.

4.3 Display Elements

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

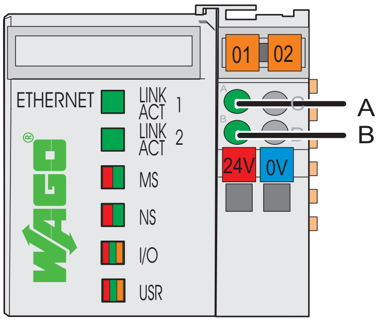


Figure 24: Display Elements

For the diagnostics of the different ranges fieldbus, node and supply voltage, the LED's can be divided into three groups:

Table 11: Display Elements Fieldbus Status

LED	Color	Meaning
LINK ACT 1	green	indicates a connection to the physical network at port 1
LINK ACT 2	green	indicates a connection to the physical network at port 2
MS	red/green	indicates the status of the node
NS	red/green	indicates the network status

Table 12: Display Elements Node Status

LED	Color	Meaning
I/O	red/green/ orange	indicates the operation of the node and signals via a blink code faults encountered
USR	red/green/ orange	indicates information to the Internal bus faults, controlled from the user programm according to the visualization programming.

Table 13: Display Elements Supply Voltage

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

Information

More information about the LED Signaling
Read the detailed description for the evaluation of the displayed LED-Signals in the chapter "Diagnostics" > "LED Signaling".

4.4 Operating Elements

4.4.1 Service Interface

The Service Interface is to find behind the flap.

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

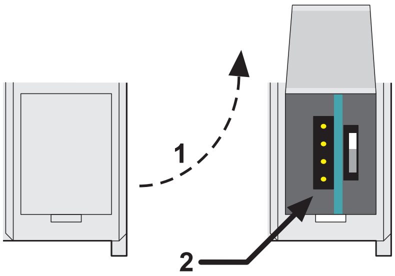


Fig. 25: Service interface for programming and configuration (closed and open door)

Table 14: Service port

Number	Description
1	Open the damper
2	Configuration and Programming Interface

NOTICE

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

The 750-920 or 750-923 Communication Cable is connected to the 4-pole header.

4.4.2 Mode Selector Switch

The mode selector switch is located behind the cover flap.

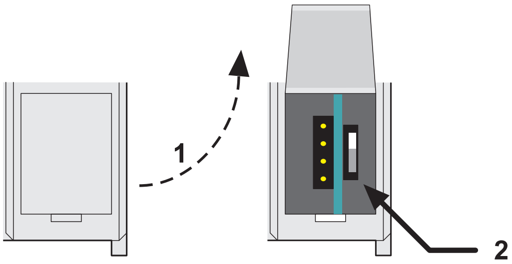


Figure 26: Mode selector switch (closed and open damper of the service port)

Table 15: Mode selector switch

Number	Description
1	Open the damper
2	Operating mode switch

The operating mode switch determines the loading, starting and stopping of the PLC-application by the controller. This multifunction sliding switch features 3 slide lock positions and a push-button function.
The sliding switch is designed for a number of operations in compliance with EN61131T2.

NOTICE

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

Note

Defining the outputs for a program stop!
In order to switch the outputs to a safe mode at the program stop, define the status of the outputs at "STOP".

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

Note

Mode selector switch position is negligible in software start/stop!
The position of the mode selector switch is not important when starting or stopping the PFC application from WAGO-I/O-*PRO*.

One of the following functions is active, depending in which of the three static positions "top", "center" or "bottom" the switch is located at a power on or in a hardware or software reset:

Table 1: Mode selector switch positions, static positions at Power On / reset

Positions of the mode selector switch	Function
Up position	"RUN" - activate program processing, Boot- project (if available) is started.
Center position	"STOP" - stop program processing, PFC- application is stopped.
Down position	After a PowerOn reset, the controller is in Bootstrap mode.

The controller performs the following functions, if a position change of the switch is performed during the current operation:

Table 2: Mode selector switch positions, dynamic positions during the current operation

Position change of the mode selector switch	Function
From the top to the center position	"STOP" - stop program processing, PFC- application is stopped.
From the center to the top position	"RUN" - activate program processing, Boot project (if available) is started.
From the center to the bottom position	No reaction. After Power On/Reset the Bootstrap loader is started on the service interface.
From the bottom to the center position	No reaction.
Press down (e.g., using a screwdriver)	Hardware reset All outputs are reset; variables are set to 0, FALSE or to an initial value. Retain variables or markers are not changed. A hardware reset can be performed either at STOP or at RUN at any position of the mode selector switch. Restart the fieldbus controller.

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

4.4.3 Address Selection Switch

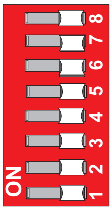


Figure 27: Address Selection Switch

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

Table 16: Meaning of DIP switch positions

Address	Meaning
0	The IP parameter is configured via the web-based management. BootP, DHCP and application of the values from the EEPROM are available to the user. In the default status, configuration via BootP is activated.
1-254	The configuration of the IP address consists of the network address (configurable and 192.168.1 by default) and the value set for the DIP switch.
255	The DHCP protocol is used to configure the IP parameters.

4.5 Technical Data

4.5.1 Device Data

Table 17: Technical data – Device data

Width	62 mm
Height (from upper-edge of DIN 35)	65 mm
Length	100 mm
Weight	approx. 160 g
Degree of protection	IP 20

4.5.2 System Data

Table 18: Technical data – System data

Number of controllers	Limited by ETHERNET specification; max. 20 x 750-881 series connected
Transmission medium	Twisted Pair S/UTP, STP 100 Ω Cat 5
Bus coupler connection	RJ-45
Max. length of fieldbus segment	100 m behind hub and 750-881
Max. length of network	2000 m
Baud rate	10/100 Mbit/s
Protocols	MODBUS/TCP (UDP), ETHERNET/IP, HTTP, BootP, DHCP, DNS, SNTP, FTP, SNMP
Programming	WAGO-I/O-*PRO*
IEC-61131-3	AWL, KOP, FUP (CFC), ST, AS
Max. number of socket links	3 HTTP, 15 MODBUS/TCP, 10 FTP, 2 SNMP, 5 for IEC-61131-3 program, 2 for WAGO-I/O-*PRO*, 128 for Ethernet/IP
Powerfail RTC Buffer	at least 6 days
Number of I/O modules - with bus extension	64 250
Configuration	via PC
Program memory	1 Mbyte
Data memory	512 kByte
Non-voltatile memory (retain)	32 kByte (16 kByte retain, 16 kByte flag)

4.5.3 Supply

Table 19: Technical data – Supply

Voltage supply	DC 24 V (-25 % ... +30 %)
Input current $_{max.}$	500 mA at 24 V
Efficiency of the power supply	90 %
Internal current consumption	450 mA at 5 V
Total current for I/O modules	1700 mA at 5 V
Isolation	500 V system/supply
Voltage via power jumper contacts	DC 24 V (-25 % ... +30 %)
Current via power jumper contacts $_{max.}$	DC 10 A

4.5.4 Fieldbus MODBUS/TCP

Table 20: Technical data – Fieldbus MODBUS/TCP

Input process image $_{max}$	2040 Byte
Output process image $_{max}$	2040 Byte
Input variables $_{max}$	512 Byte
Output variables $_{max}$	512 Byte

4.5.5 Accessories

Table 21: Technical data – Accessories

Miniature WSB Quick marking system
WAGO-I/O-*PRO*

4.5.6 Wire Connection

Table 22: Technical Data Wire Connection

Wire connection	CAGE CLAMP®
Cross section	0.08 mm² ... 2.5 mm², AWG 28-14
Stripped lengths	8 ... 9 mm / 0.33 in
Power jumper contacts	blade/spring contact, self-cleaning
Voltage drop at $I_{max.}$	< 1 V/64 modules
Data contacts	slide contact, hard gold plated 1.5 µm, self-cleaning

4.5.7 Climatic environmental conditions

Table 23: Technical Data - Climatic environmental conditions

Operating temperature range	0 °C ... 55 °C
Storage temperature range	-25 °C ... +85 °C
Relative humidity without condensation	max. 95 %
Resistance to harmful substances	Acc. to IEC 60068-2-42 and IEC 60068-2-43
Maximum pollutant concentration at relative humidity < 75%	$SO_2 \leq 25$ ppm $H_2S \leq 10$ ppm
Special conditions	Ensure that additional measures for components are taken, which are used in an environment involving: – dust, caustic vapors or gases – ionizing radiation

NOTICE

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

4.5.8 Mechanical strength

Table 24: Technical data – Mechanical strength

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

4.6 Approvals

Information

More Information about Approvals

Detailed references to the approvals are listed in the document "Overview Approvals **WAGO-I/O-SYSTEM 750**", which you can find on the DVD "AUTOMATION Tools and Docs" (order no. 0888-0412) or via the internet under: www.wago.com → Documentation → WAGO-I/O-SYSTEM 750 → System Description.

The following approvals have been granted to 750-881 fieldbus coupler/controller:

CE	Conformity Marking
cULus	$_C$UL$_{US}$ (UL508)

The following Ex approvals have been granted to 750-881 fieldbus coupler/controller:

TÜV 07 ATEX 554086 X

I M2 Ex db I Mb
II 3 G Ex nAc IIC T4 Gc
II 3 D Ex tc IIIC T135°C Dc

Permissible operation temperature: 0 °C ≤ T_A ≤ +60 °C

TÜV TUN 09.0001X

Ex db I Mb
Ex nAc IIC T4 Gc
Ex tc IIIC T135°C Dc

Permissible operation temperature: 0 °C ≤ T_A ≤ +60 °C

The following ship approvals are pending for 750-881 fieldbus coupler/controller:

GL	GL (Germanischer Lloyd)	Cat. A, B, C, D (EMC 1)

Information

For more information about the ship approvals:

Note the "Supplementary Power Supply Regulations" chapter for the ship approvals.

4.7 Standards and Guidelines

750-881 meets the following requirements on emission and immunity of interference:

EMC CE-Immunity to interference	acc. to EN 61000-6-2: 2005
EMC CE-Emission of interference	acc. to EN 61000-6-3: 2007
EMC marine applications-Immunity to interference	acc. to Germanischer Lloyd (2003)
EMC marine applications-Emission of interference	acc. to Germanischer Lloyd (2003)

5 Assembly

5.1 Installation Position

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

Note

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

WAGO order no. 249-116	End stop for DIN 35 rail, 6 mm wide
WAGO order no. 249-117	End stop for DIN 35 rail, 10 mm wide

5.2 Total Extension

The length of the module assembly (including one end module of 12mm width) that can be connected to the 750-881 is 780 mm. When assembled, the I/O modules have a maximum length of 768 mm.

Examples:

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

Exception:

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

NOTICE

Observe maximum total length of a node!
The maximum total length of a node without a coupler/controller must not exceed 780 mm. Furthermore, you must observe restrictions made on certain types of couplers/controllers.

Note

Increase total length using a WAGO internal data bus extension module!
Using an internal data bus extension module from WAGO, you can increase the total length of the fieldbus node. In this type of configuration, you must connect a 750-627 Bus Extension End Module to the last module of the node.
You then connect the 750-627 module to the 750-628 Coupler Module of the next I/O module assembly via RJ-45 cable.
You can connect up to 10 internal data bus extension coupler modules 750-628 to an internal data bus extension end module 750-627. In this manner, you can logically connect up to 10 module assemblies to a 750-881, dividing a fieldbus node into 11 assemblies maximum.
The maximum cable length between two assemblies is 5 meters. For additional information, refer to the "750-627/-628 Modules" manual. The total cable length for a fieldbus node is 70 meters.

5.3 Assembly onto Carrier Rail

5.3.1 Carrier Rail Properties

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

NOTICE

Do not use any third-party carrier rails without approval by WAGO!
WAGO Kontakttechnik GmbH & Co. KG supplies standardized carrier rails that are optimal for use with the I/O system. If other carrier rails are used, then a technical inspection and approval of the rail by WAGO Kontakttechnik GmbH & Co. KG should take place.

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

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

5.3.2 WAGO DIN Rail

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

Table 25: WAGO DIN Rail

Order number	Description
210-113 /-112	35 x 7,5; 1 mm; steel yellow chromated; slotted/unslotted
210-114 /-197	35 x 15; 1,5 mm; steel yellow chromated; slotted/unslotted
210-118	35 x 15; 2,3 mm; steel yellow chromated; unslotted
210-198	35 x 15; 2,3 mm; copper; unslotted
210-196	35 x 7,5; 1 mm; aluminum; unslotted

5.4 Spacing

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

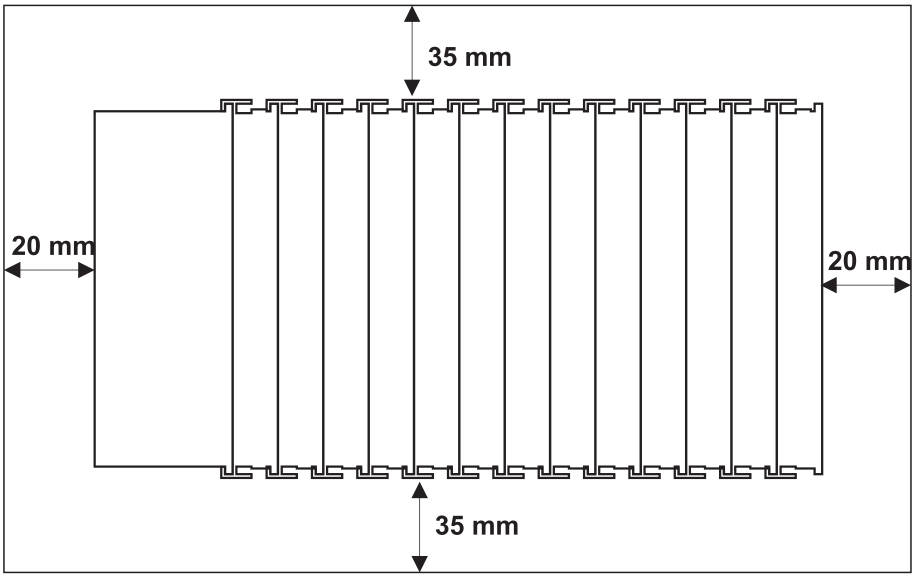


Figure 28: Spacing

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

5.5 Assembly Sequence

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

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

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

⚠ CAUTION

Risk of injury due to sharp-edged male contacts!
The male contacts are sharp-edged. Handle the module carefully to prevent injury.

NOTICE

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

NOTICE

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

Note

Don't forget the bus end module!
Always plug a bus end module 750-600 onto the end of the fieldbus node! You must always use a bus end module at all fieldbus nodes with the WAGO I/O System 750 fieldbus couplers/controllers to guarantee proper data transfer.

5.6 Inserting and Removing Devices

DANGER

Use caution when interrupting the PE!
Make sure that people or equipment are not placed at risk when removing an I/O module and the associated PE interruption. To prevent interruptions, provide ring feeding of the ground conductor, see section "Grounding/Ground Conductor" in manual "System Description WAGO-I/O-SYSTEM 750".

NOTICE

Perform work on devices only if the system is de-energized!
Working on devices when the system is energized can damage the devices. Therefore, turn off the power supply before working on the devices.

5.6.1 Inserting the Fieldbus Coupler/Controller

1. When replacing the fieldbus coupler/controller for an already available fieldbus coupler/controller, position the new fieldbus coupler/controller so that the tongue and groove joints to the subsequent I/O module are engaged.

2. Snap the fieldbus coupler/controller onto the carrier rail.

3. Use a screwdriver blade to turn the locking disc until the nose of the locking disc engages behind the carrier rail (see the following figure). This prevents the fieldbus coupler/controller from canting on the carrier rail.

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

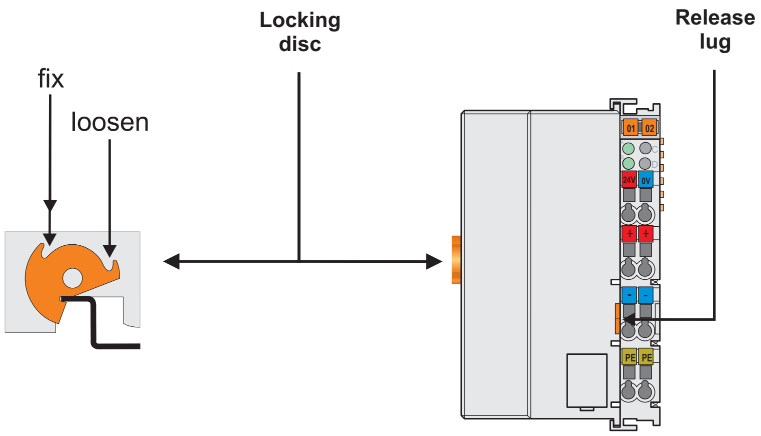


Figure 29: Unlocking lug of extended ECO coupler

5.6.2 Removing the Fieldbus Coupler/Controller

1. Use a screwdriver blade to turn the locking disc until the nose of the locking disc no longer engages behind the carrier rail.

2. Remove the fieldbus coupler/controller from the assembly by pulling the release tab.

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

5.6.3 Inserting I/O Module

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

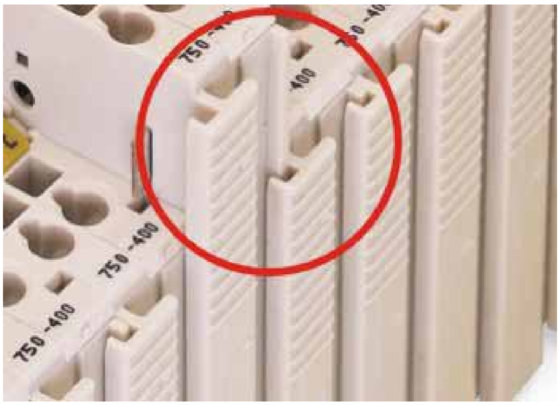


Figure 30: Insert I/O module

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

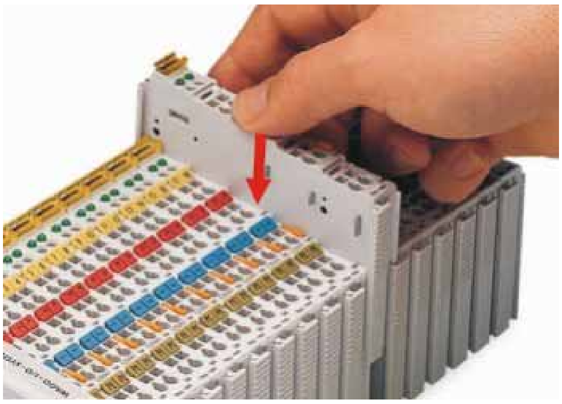


Figure 31: Snap the I/O module into place

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

5.6.4 Removing the I/O Module

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

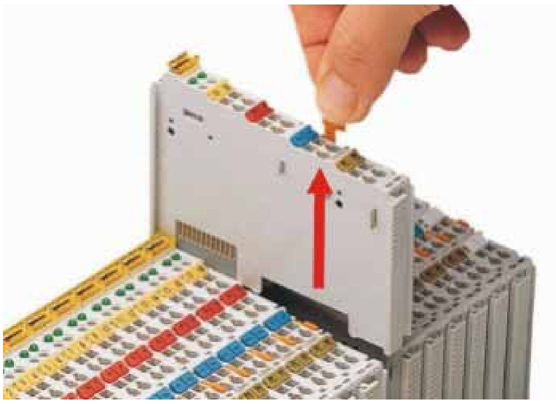


Figure 32: Removing the I/O module

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

6 Connect Devices

6.1 Data Contacts/Internal Bus

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

Figure 33: Data contacts

NOTICE

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

NOTICE

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

6.2 Power Contacts/Field Supply

⚠ CAUTION

Risk of injury due to sharp-edged male contacts!
The male contacts are sharp-edged. Handle the module carefully to prevent injury.

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

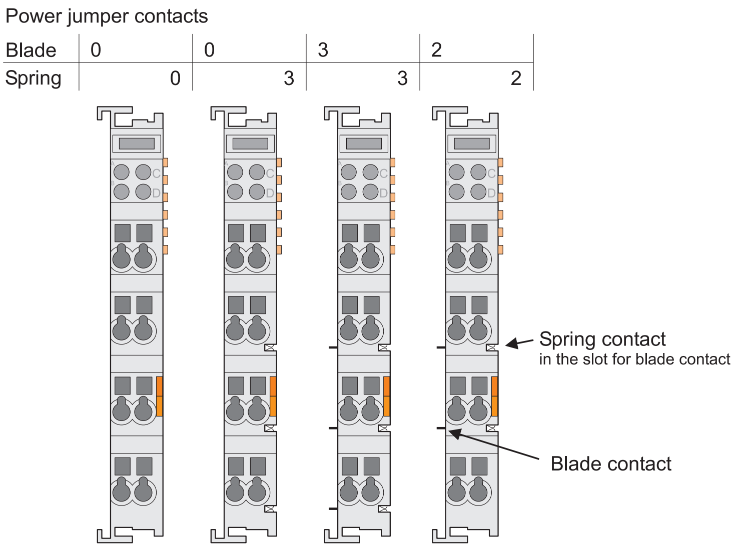


Figure 34: Example for the arrangement of power contacts

Note

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

6.3 Connecting a conductor to the CAGE CLAMP®

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

Note

Only connect one conductor to each CAGE CLAMP® connection!
Only one conductor may be connected to each CAGE CLAMP® connection.
Do not connect more than one conductor at one single connection!

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

Exception:
If it is unavoidable to jointly connect 2 conductors, then you must use a ferrule to join the wires together. The following ferrules can be used:

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

1. To open the CAGE CLAMP® insert the actuating tool into the opening above the connection.
2. Insert the conductor into the corresponding connection opening.
3. To close the CAGE CLAMP® simply remove the tool - the conductor is then clamped firmly in place.

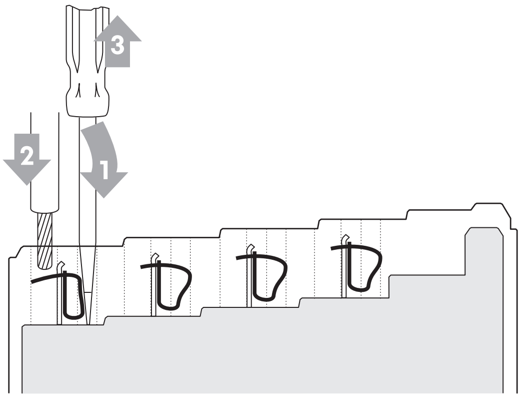


Figure 35: Connecting a conductor to a CAGE CLAMP®

7 Function Description

7.1 Operating System

7.1.1 Run-up

Note

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

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

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

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

7.1.2 PFC Cycle

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

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

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

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

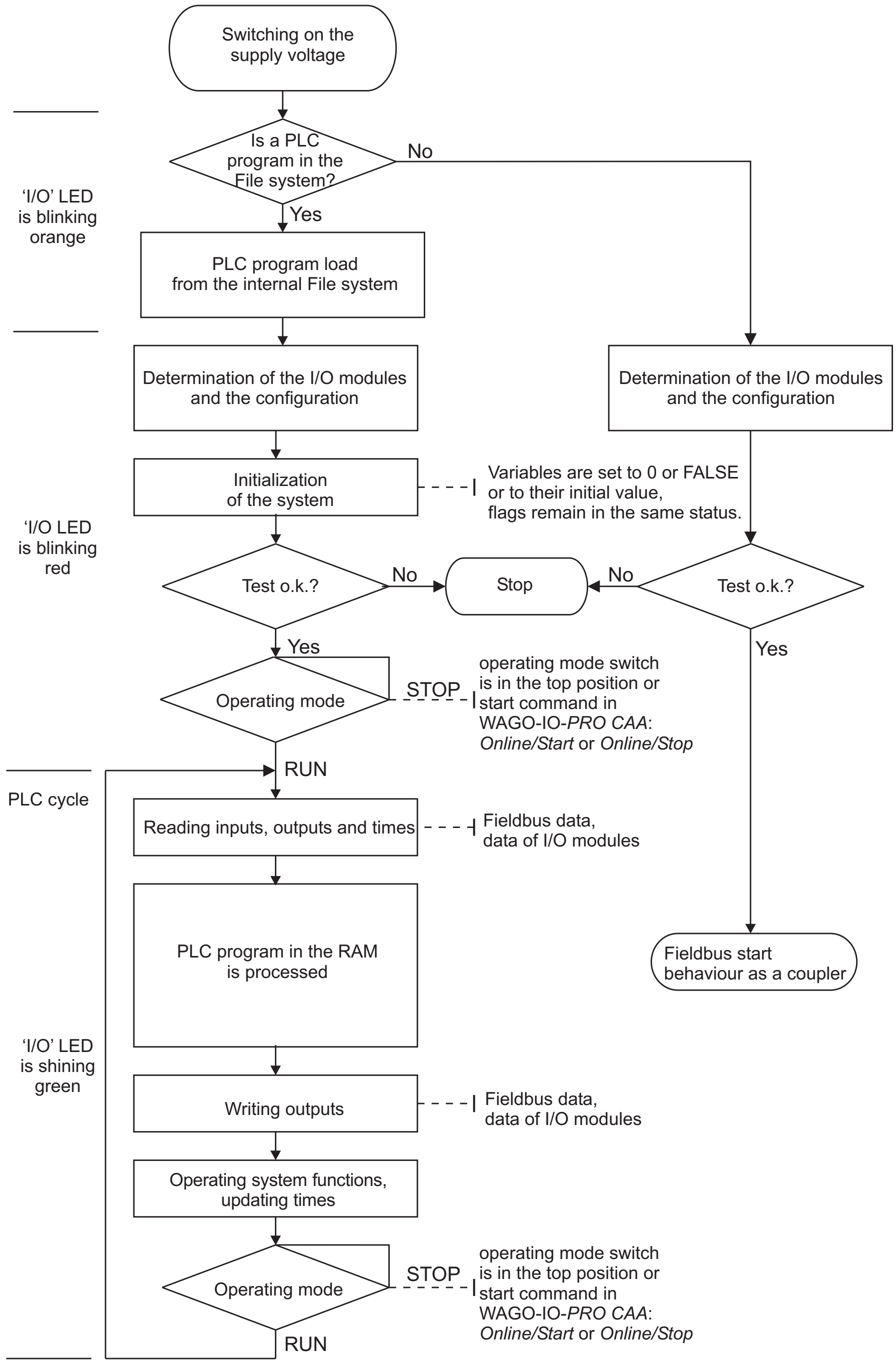


Figure 36: Run-up of the Controller

7.2 Process Data Architecture

7.2.1 Basic Structure

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

Note

Up to 250 I/O modules can be connected with the data bus extension modules.
Using the WAGO module bus extension coupler module 750-628 and end module 750-627 makes it possible to connect up to 250 modules to the Programmable Fieldbus Controller ETHERNET.

Information

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

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

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

This group includes: counter modules, angle and distance measurement modules and communication modules.

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

Note

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

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

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

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

The Ethernet/IP PFC variables are then mapped behind the remaining physical I/O module data.This memory range includes words 1276 ... 1531.

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

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

For the fieldbus controller, a MODBUS/TCP master can access the data via implemented MODBUS functions, whereby decimal or hexadecimal MODBUS addresses are used.
Optionally, data can also be accessed via Ethernet/IP using an object model.

Information

Additional Information:
For a detailed description of these fieldbus-specific data access methods, refer to the section "MODBUS Functions" or the section "Ethernet/IP (Ethernet/Industrial Protocol)".

Information

Additional Information:
For the fieldbus-specific process image of any WAGO I/O module, please refer to the section "Structure of the process data".

7.2.2 Example of an Input Process Image

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

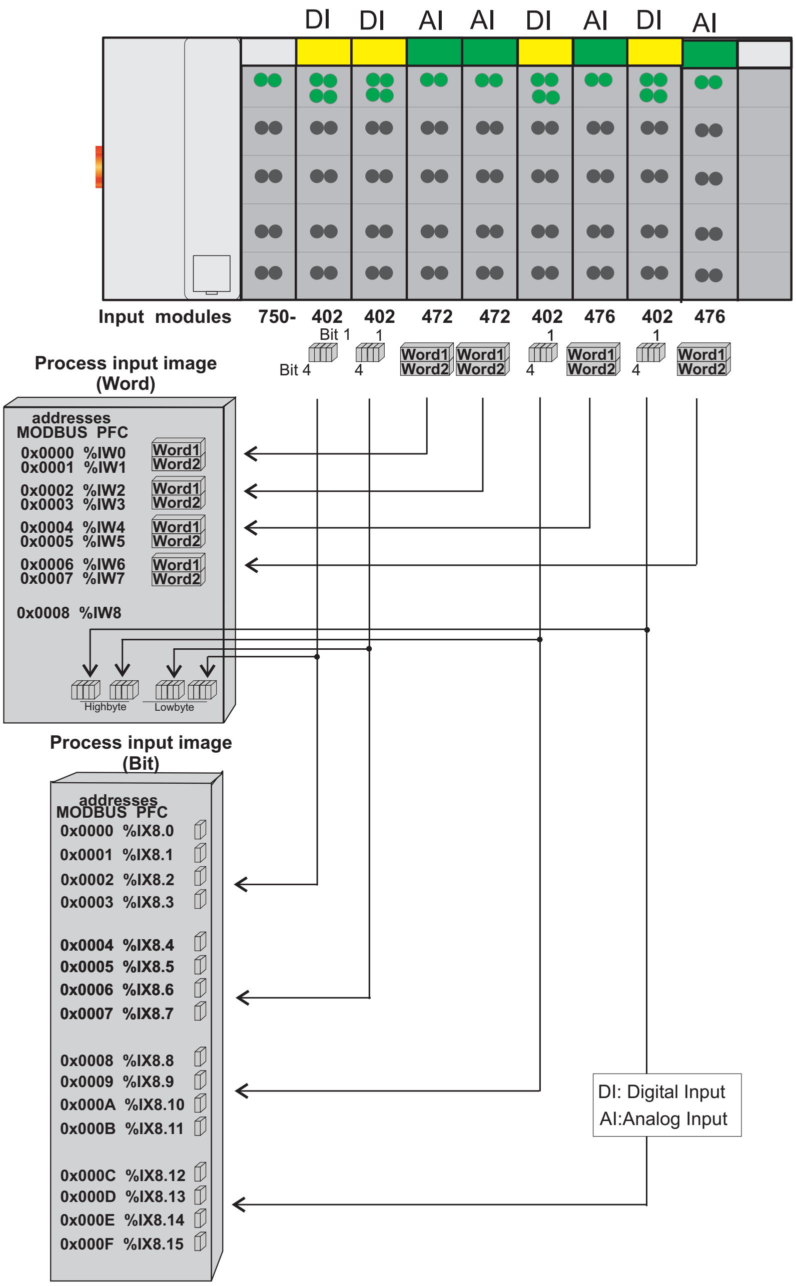


Figure 37: Example of process image for input data

7.2.3 Example of an Output Data Process Image

The following example for the output process image comprises 2 digital and 4 analog outputs. It comprises 4 words for the analog outputs and 1 word for the digital outputs; i.e., 5 words in total.
In addition, the output data can also be read back with an offset of 200_{hex} (0x0200) added to the MODBUS address.

Note

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

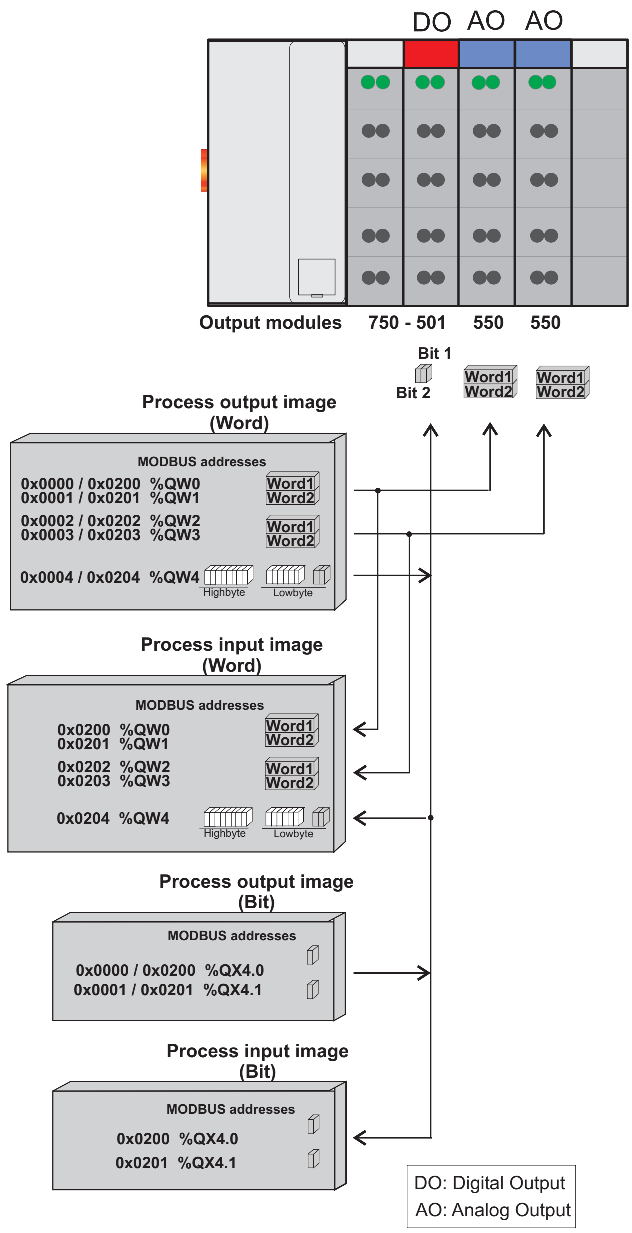


Figure 38: Example of process image for output data

7.2.4 Process Data MODBUS/TCP and EtherNet/IP

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

For the fieldbus controller with MODBUS and Ethernet/IP, the process image is built up word-by-word (with word alignment). The internal mapping method for data greater than one byte conforms to Intel formats.

Information

Additional Information:
For the respective fieldbus-specific structure of the process values of any I/O module within the 750 or 753 Series of the WAGO-I/O-SYSTEM, refer to Section "Structure of Process Data for MODBUS/TCP" or "Structure of Process Data for Ethernet/IP".

7.3 Data Exchange

With the fieldbus controller, data is exchanged via the MODBUS/TCP protocol and/or the MODBUS/UDP protocol or Ethernet/IP.

MODBUS/TCP works according to the master/slave principle. The master controller can be a PC or a PLC.

The fieldbus controllers of the WAGO-I/O-SYSTEM 750 are usually slave devices. Thanks to the programming with IEC 61131-3, however, these controllers can also assume the master function.

The master requests communication. This request can be directed to certain nodes by addressing. The nodes receive the request and, depending on the request type, send a reply to the master.

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

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

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

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

Data exchange takes place between the fieldbus master and the I/O modules, between the PLC function of the PFC (CPU) and the I/O modules and between the fieldbus master and the PLC function of the PFC (CPU).

If MODBUS is used as the fieldbus, the MODBUS master accesses the date using the MODBUS functions implemented in the controller; Ethernet/IP, in contrast, uses an object model for data access.

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

7.3.1 Memory Areas

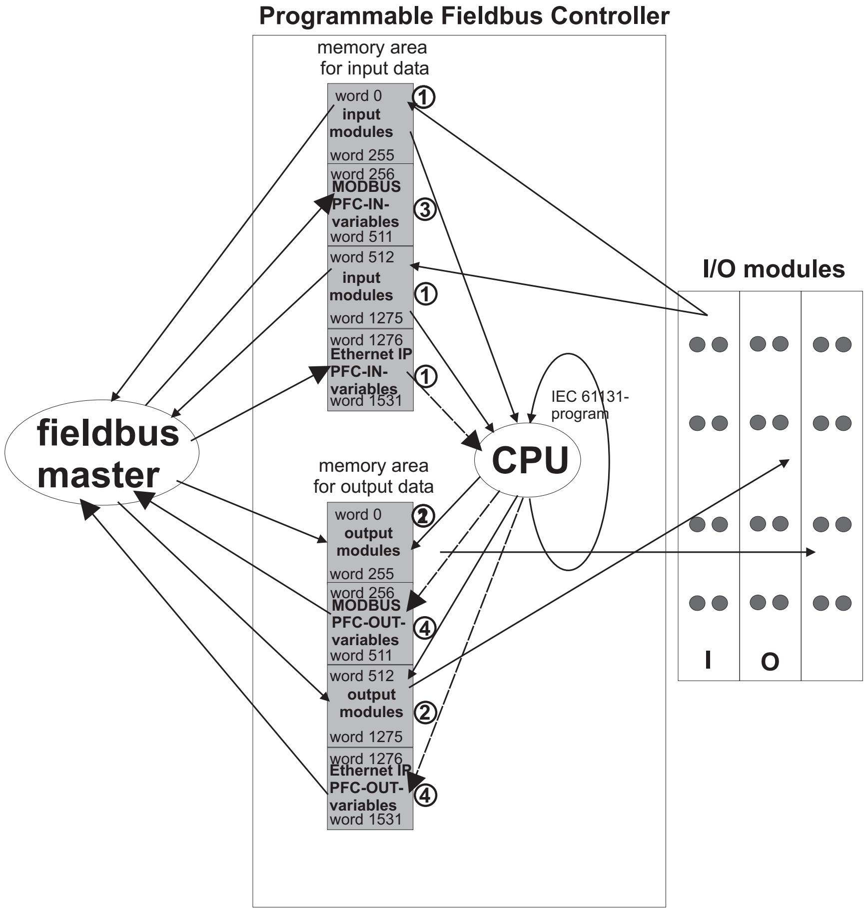


Figure 39: Memory areas and data exchange

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

① The input module data can be read by the CPU and by the fieldbus side.

② Likewise, data can be written to the output modules from the CPU and the fieldbus side.

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

③ The MODBUS-PFC input variables are written to the input memory area from the fieldbus side and read in by the CPU for processing.

④ The variables processed by the CPU using the IEC-61131-3 program are places in the output memory area, where they can be read out by the master.

The memory area for word 1276 ... 1531 for the Ethernet/IP PFC variables is adjacent to the physical I/O module data.

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

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

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

- **Data memory (1024 kByte)**
  The data memory is a volatile RAM memory for creating variables that are not required for communication with the interfaces, but rather for internal processing procedures, such as calculation of results.

- **Program memory (1024 kByte)**
  The IEC-61131-3 program is stored in the program memory. The code memory is a Flash ROM. When power is switched on, the program is transferred from the flash to the RAM memory. After error-free run-up, the PFC cycle starts with the mode selector switch at the top position, or on the Start command from the WAGO-I/O-PRO.

- **NOVRAM Remanent memory (32 kByte)**
  The remanent memory is a non-volatile memory; i.e., all values of flags and variables, that are explicitly defined by “var retain”, are retained even after a loss of power. Memory management is performed automatically. The 32 kByte memory area is normally divided into an 16 kByte addressable range for flags (%MW0 ... %MW 8191) and a 16 kByte retain area for variables without memory area addressing, that are defined by "var retain".

Note

Markers are only remanent under "var retain“!
Please note that the bit memory is only retentive if you have declared it as such under "var retain".

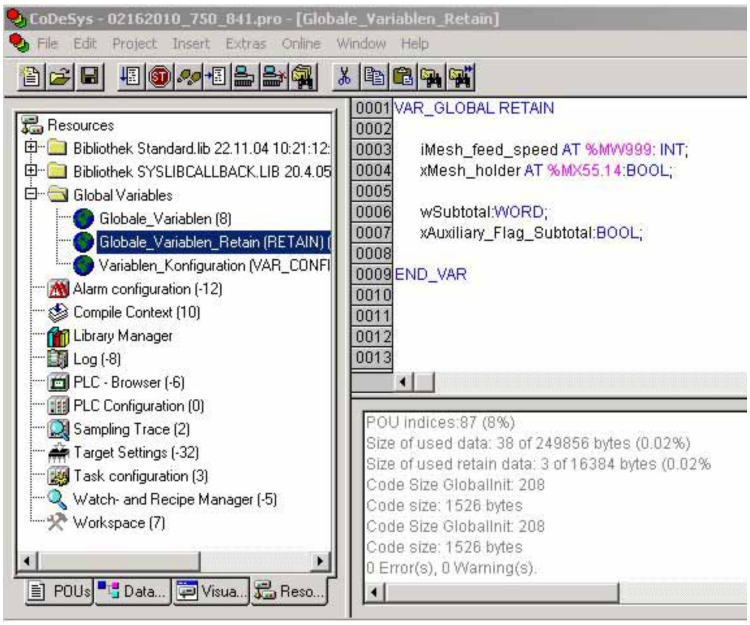


Figure 40: Example declaration of remanent flags by „var retain“

This breakdown can be varied (see following explanation).

Note

NOVRAM memory allocation can be changed in WAGO-I/O-PRO!
The breakdown of the NOVRAM can be modified when required in the programming software WAGO-I/O-*PRO*/Register "Resources"/Dialog window "Target system settings".
The start address for the flag area is fixed. The area sizes and the start address for the retain memory can be varied.
We do recommend keeping the standard settings, however, in order to avoid any overlapping of the areas.
In these default settings the size of the flag area is set at 16#4000, followed by the retain memory, with the size 16#4000

7.3.2 Addressing

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

Note

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

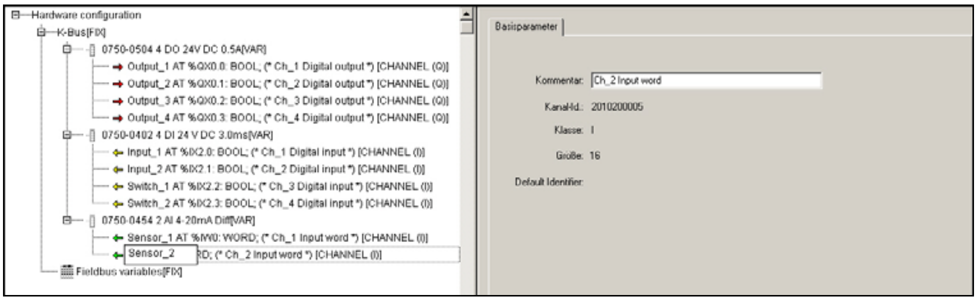


Figure 41: WAGO I/O Configurator

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

7.3.2.1 Addressing of I/O Modules

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

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

Note

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

Note

Observe process data quantity!
For the number of input and output bits or bytes of the individual IO modules please refer to the corresponding description of the IO modules.

Table 26: Data with for I/O modules

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

7.3.2.2 Address Ranges

Subdivision of the address ranges for word-by-word addressing in accordance with IEC-61131-3:

Table 27: Breakdown of address range

Word	Data
0-255	Physical I/O modules
256-511	MODBUS PFC variables
512-1275	Other physical I/O modules
1276-1531	Ethernet/IP PFC variables
1532-...	Reserved for PFC variables with future protocols

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

Table 28: Address range Word 0...255

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

Word 256...511: Address range for MODBUS-PFC variables:

Table 29: Address range, word 256 – 511

Data width	**Address**								
Bit	256..0 ... 256.7	256..8 ... 256.15	257.0 ... 257.7	257.8 ... 257.15	...	510.0 ... 510.7	510.8 ... 510.15	511.0 ... 511.7	511.8 ... 511.15
Byte	512	513	514	515	...	1020	1021	1022	1023
Word	**256**		**257**		**...**	**510**		**511**	
DWord	128				...	255			

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

Table 30: Address range, word 512 - 1275

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

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

Table 31: Address range, word 1276...1531

Data width	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			

Address range for flags:

Table 32: Address range for flags

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

IEC-61131-3 Overview of Address Areas:

Table 33: IEC-61131-3 address areas

Address area	MODBUS Access	PLC Access	Description
phys. inputs	read	read	Physical inputs (%IW0...%IW255 und %IW512...%IW1275)
phys. outputs	read/write	read/write	Physical outputs (%QW0...%QW255 und %QW512...%QW1275)
MODBUS/TCP PFC-IN variables	read/write	read	Volatile PLC input variables (%IW256...%IW511)
MODBUS/TCP PFC-OUT variables	read	read/write	Volatile PLC output variables (%QW256...%QW511)
Ethernet/IP PFC-IN variables	-	read	Volatile PLC input variables (%IW1276 ... %IW1531)
Ethernet/IP PFC-OUT variables	-	read/write	Volatile PLC output variables (%QW1276 ... %QW1531)
Configuration register	read/write	-	see Section "MODBUS Functions → MODBUS Registers → Configuration Registers"
Firmware register	read	-	see Section "MODBUS Functions → MODBUS Registers → Firmware Information Registers"
Retain variables	read/write	read/write	Remanent memory (%MW0...%MW8192)

7.3.2.3 Absolute Addressing

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

Table 34: Absolute Addressing

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

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

* The designator "X" for bits can be omitted

Note

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

Addressing example:

Table 35: Addressing example

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

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

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

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

Bit address: Word address .0 to .15
Byte address: 1st byte: 2 x word address
2nd byte: 2 x word address + 1

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

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

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

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

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

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

Table 36: Allocation of digital inputs and outputs to process data words in accordance with the Intel format

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

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

Note

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

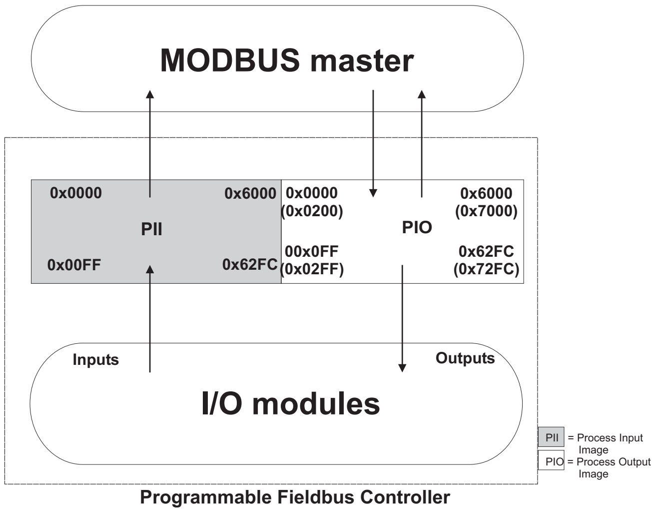


Figure 42: Data exchange between MODBUS Master and I/O modules

Register functions start at address 0x1000. These functions can be addressed in a similar manner with the MODBUS function codes that are implemented (read/write).
The specific register address is then specified instead of the address for a module channel.

Information

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

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

The data exchange between Ethernet/IP master and the I/O modules is objectoriented. Each node on the network is depicted as a collection of objects.

The "assembly" object specifies the structure of the objects for the data transmission. With the assembly object, data (e.g. I/O data) can be combined into blocks (mapped) and sent via a single message connection. Thanks to this mapping, less access to the network is necessary.

There is a distinction between input and output assemblies.

An input assembly reads in data from the application via the network or produces data on the network.

An output assembly writes data to the application or consumes data from the network.

In the fieldbus coupler/controller, various assembly instances are already preprogrammed (static assembly).

After the input voltage is applied, the assembly object combines data from the process image. As soon as a connection is established, the master can address the data with "class", "instance", and "attribute" and access it or read and write using I/O connections.

The mapping of the data depends on the assembly instance of the static assembly selected.

Information

Additional Information:
The assembly instances for the static assembly are described in the section "Ethernet/IP".

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

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

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

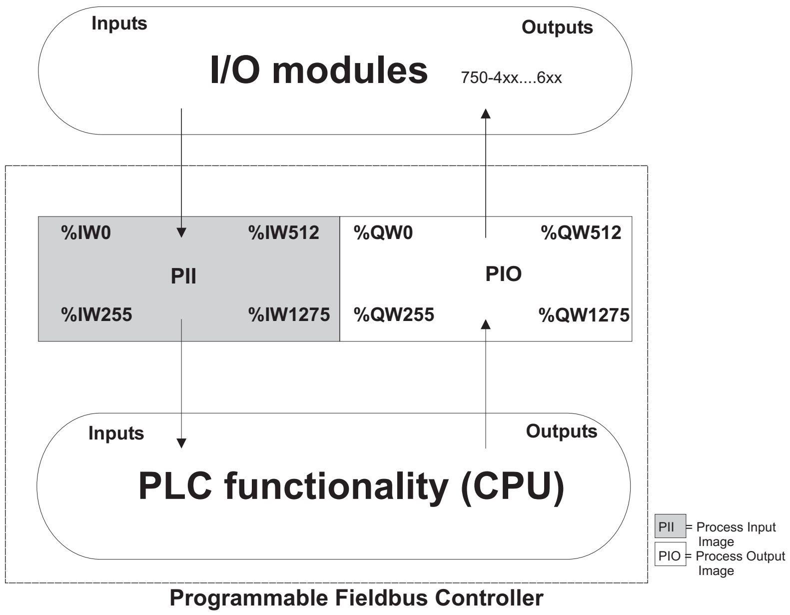


Figure 43: Data exchange between PLC function (CPU) of the PFC and the I/O modules

7.3.5 Data Exchange between Master and PLC Function (CPU)

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

Variable data generated by the master are routed as input variables to the PFC, where they are further processed.
Data created in the PFC are transmitted via fieldbus to the master as output variables.

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

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

Data access by the MODBUS/TCP Master

Access to data by the MODBUS Master is always either by word or by bit.
Addressing of the first 256 data words by the I/O modules begins with word-by-word and bit-by-bit access at 0.
Addressing of the data by the variables begins at 256 for word-based access; bit-by-bit access then takes place starting at:

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

The bit number can be determined using the following equation:

Bit No. = (word * 16) + Bit No. in word
Example: 4097 = (256 * 16) + 1

Data Access by PLC Function (CPU)

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

Example:

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

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

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

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

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

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

Information

Additional Information
There is a detailed description of the MODBUS and the corresponding IEC 61131 addressing in section "MODBUS Register Mapping".

7.3.6 Application Example

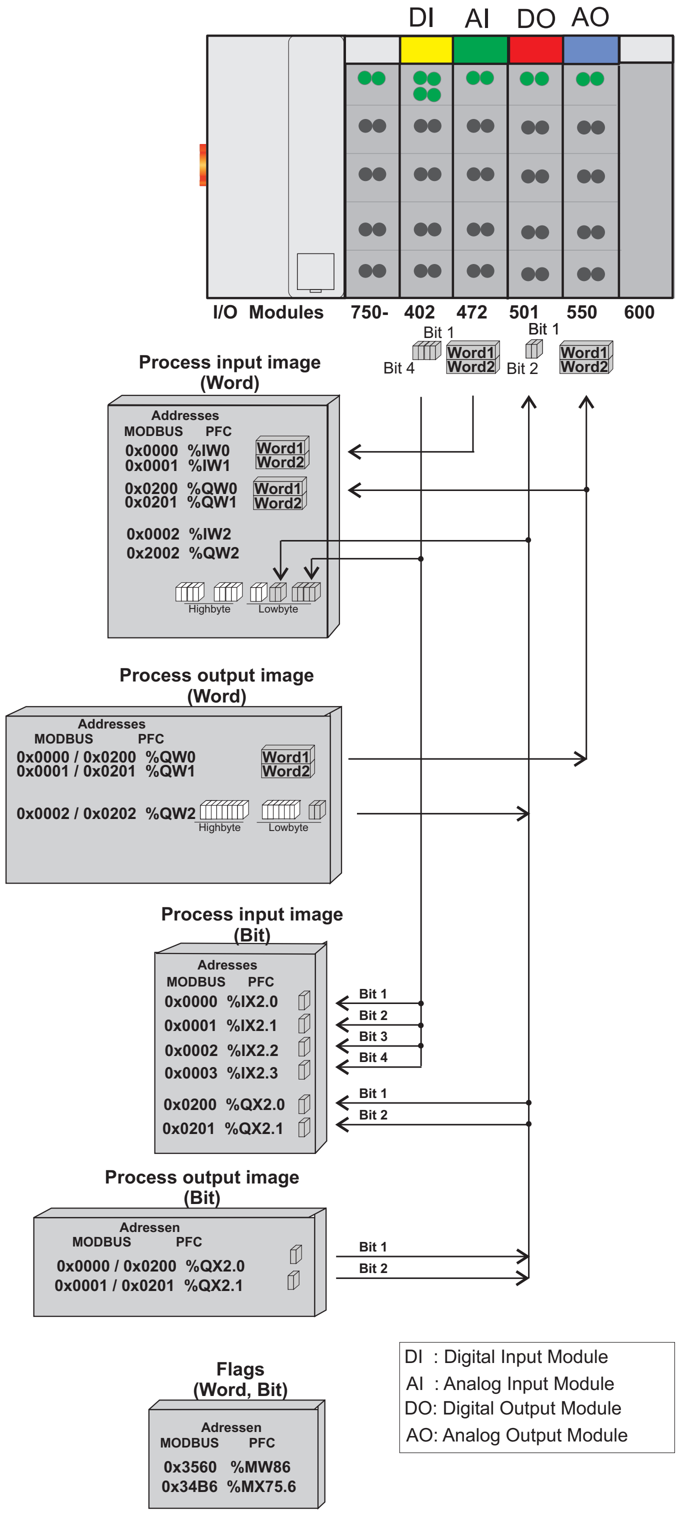


Figure 44: Example of addressing for a fieldbus node

8 Commissioning

This chapter shows a step-by-step procedure for starting up exemplariliy a WAGO fieldbus node.

Note

Good example!
This description is just an example and only serves to describe the procedure for a local start-up of a single fieldbus node with a non-networked computer under Windows.

Two work steps are required for start-up. The description of these work steps can be found in the corresponding following sections.

- **Connecting client PC and fieldbus nodes**
- **Assigning the IP address to the fieldbus node**

Note

The IP address must occur in the network only once!
For error-free network communication, note that the assigned IP address must occur only once in the network!
In the event of an error, the error message "IP address configuration error" (error code 6 - error argument 6) is indicated by 'I/O' LED at the next power-on.

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

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

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

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

8.1 Connecting Client PC and Fieldbus Nodes

1. Mount the fieldbus to the carrier rail.
   Observe the installation instructions described in "Assembly" section.

2. Connect the 24V power supply to the supply module.

3. Connect an Ethernet interface from the client PC to an Ethernet interface of the fieldbus controller

4. Turn the operating voltage on.
   Make sure that the mode selector is not in the bottom position.

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

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

8.2 Allocating the IP Address to the Fieldbus Node

- Use **address selection switch** (DIP switch) to assign IP address (manually).
- **Automatic assignment of addresses via DHCP**
- **Assigning IP Address via BootP server**

8.2.1 Assigning IP Address via Address Selection Switch

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

Example:

DIP switch IP address:	192.168.7
Set DIP switch value:	**50** (binary coded: 00110010)
Resulting IP address saved in the fieldbus controller:	192.168.7.**50**

Note

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

Note

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

The base address used consists of the first three bytes of the IP address. This always depends on the DIP switch IP address currently saved in the fieldbus controller.
If there are still no static IP address in the fieldbus controller, the default value **192.168.1** defined by the firmware as the base address is used when setting the DIP switch to 1 - 254.
The address selection switch setting then overwrites the value of the host ID.

Information

More information about changing the static base address
You can also change the base address currently saved in the fieldbus controller as required.
Either proceed as described in the following section "Assigning IP Address via Web Server".

1. To configure the IP address via the address selection switch by setting the host ID (last position of the IP address) to a value that does not equal 0/255, first convert the host ID to the binary representation.
   For example, host ID 50 results in a binary code of 00110010.

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

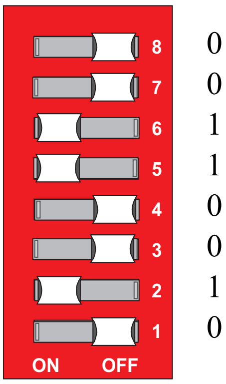


Figure 45: Address selection switch

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

8.2.2 Assigning IP Address via DHCP

If you want to use DHCP to assign the IP address, it happens automatically via a DHCP server on the network.

Note

Total network failure when there are two DHCP servers in the network!
To prevent network failure, never connect a PC, on which a DHCP server is installed, to a global network. In larger networks, there is usually a DHCP server already that can cause collisions and subsequent network failure.

Note

There must be a DHCP server in the network for further configuration!
Install a DHCP server on your client PC in the local network if not already available. You can download a DHCP server free of charge on the Internet, e.g., http://windowspedia.de/dhcp-server_download/.

Note

Assign the client PC a fixed IP address and note common subnet!
Note that the client PC, on which the DHCP server is listed, must have a fixed IP address and that the fieldbus node and client PC must be in the same subnet.

The following steps are included:

- Enable DHCP
- Disable DHCP

8.2.2.1 Enable DHCP

Note

Set the address selection switch to 255 for active software configuration!
Set the address selection switch to 255 to disable the DIP switch and to enable DHCP.
Restart the fieldbus node after adjusting the address selection switch to apply the configuration changes.

8.2.2.2 Disabling DHCP

Note

BootP must be disabled to assign the address permanently!
To apply the new IP address permanently in the fieldbus controller, BootP must be disabled.
This prevents the fieldbus coupler from receiving a new BootP request.

You can disable DHCP in two ways:

- Disable DHCP via the address selection switch.
- Disable DHCP in the Web-based Management System.

Disable DHCP via the address selection switch.

Note

Do not set the address selection switch to 0/255 again!
Do not switch the address selection switch to 0/255 again because doing so automatically disables the DIP switch and enables IP address assignment via the software configuration.

1. Use the address selection switch to set a value between 1 ... 254 and the DIP switch IP address saved in the fieldbus controller (with changed Host ID = DIP switch) is then valid.
(Example: If the DIP switch IP address 10.127.3 was saved in the fieldbus controller and you set the switch to 50 (binary coded 00110010), for example, the fieldbus controller then has the address 10.127.3.50.)

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

Disable DHCP in the Web-based Management System

Note

Set the address selection switch to 0 for active software configuration!
Set the address selection switch to 0 to disable address selection via DIP switch or DHCP.

1. Set the address selection switch to 0.

2. Launch a Web browser (e.g., MS Internet Explorer or Mozilla) and enter the IP address you have assigned your fieldbus node in the address bar.

3. Click **[Enter]** to confirm.
   The start page of the Web based Management System loads.

4. Select "Port" in the left menu bar.

5. Enter your user name and password in the inquiry screen (default: user = "admin", password = "wago" or user = "user", password = "user").
   The HTML page "Port configuration" loads:

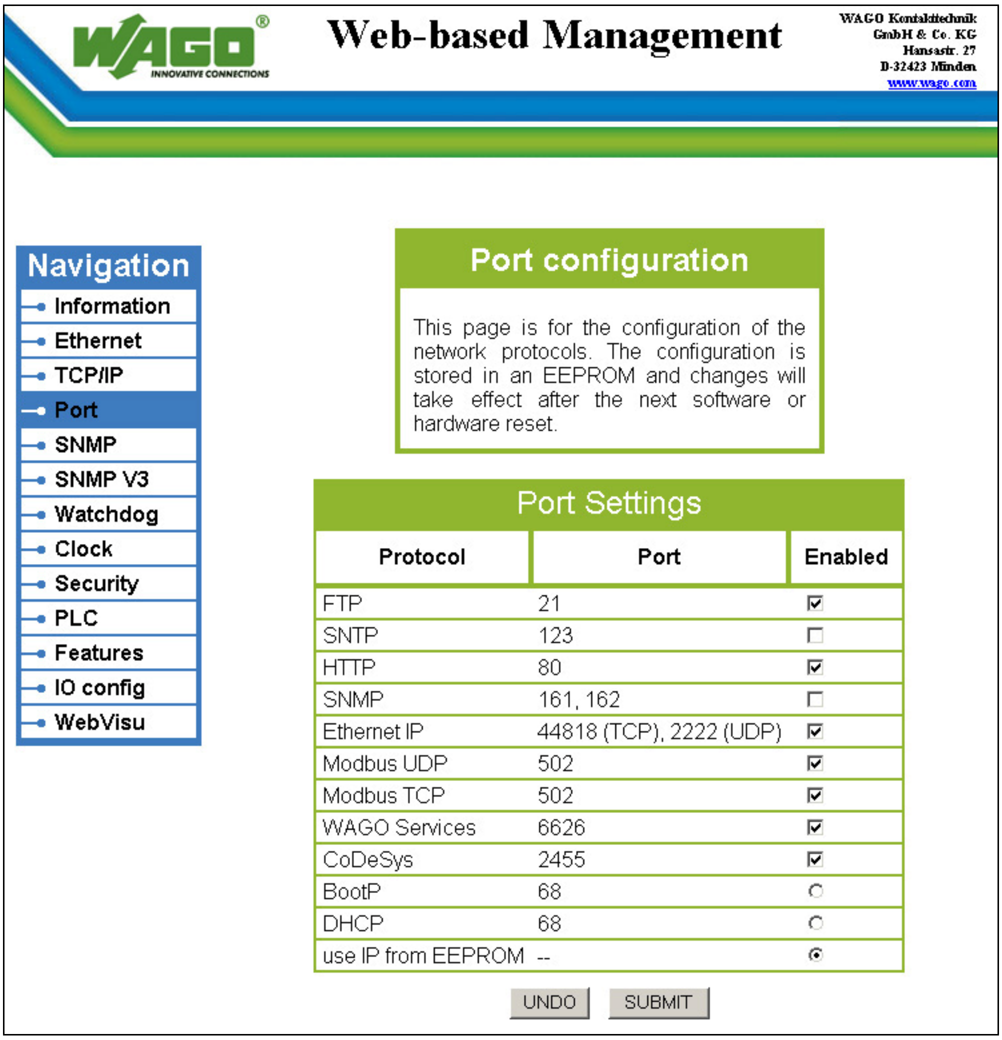


Figure 46: WBM page "Port"

6. Disable DHCP by selecting the option “**BootP**” or “**use IP fom EEPROM**”.

7. Click on **[SUBMIT]** to apply the changes in your fieldbus node.

8. Restart the fieldbus node to apply the settings of the Web interface.

8.2.3 Assigning the IP Address with a BootP Server

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

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

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

Note

Set the address selection switch to 0 for active software configuration!
Set the address selection switch to 0 to disable the DIP switch and to enable the software configuration via BootP.
Restart the fieldbus node after adjusting the address selection switch to apply the configuration changes.

Note

IP address assignment is not possible via the router!
The IP address is assigned via patch cable, switches, hubs, or via direct link using a crossover cable. Addresses can not be allocated via router.

Note

BootP must be enabled on the Web pages!
Note that BootP must be enabled on the internal Web pages of the WBM, HTML page "Port configuration".
BootP is enabled by default when delivered.

Information

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

Information

More information about the WAGO-BootP-Server
The "WAGO-BootP-Server 759-315" is available free of charge on the CD "AUTOMATION Tools and Docs" (Art. No.: 0888-0412) or at http://www.wago.com under Downloads → AUTOMATION → 759-315 WAGO-BootP-Server.

The following steps are included:

- Note MAC ID
- Note IP address
- Assigning the IP address and enable BootP
- Disable BootP

8.2.3.1 Note MAC ID

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

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

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

2. Plug the fieldbus controller into the assembly of the fieldbus node.

3. Use the fieldbus cable to connect the fieldbus connection of your mechanically and electrically assembled fieldbus node to an open interface on your computer.
   The client PC must be equipped with a network card for this connection.
   The controller transfer rate then depends on the network card of your client PC.

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

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

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

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

Information

More information about LED signaling
The exact description for evaluating the LED signal displayed is available in the section "Diagnostics", "LED Signaling".

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

8.2.3.2 Determining IP addresses

1. If the client PC is already integrated into an IP network, you can determine the client PC's IP address by clicking on **Control Panel** from the **Start Menu / Settings**.

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

   For Windows NT:

   - Select the **Protocols** tab
   - Mark the entry TCP/IP protocol

   For Windows 2000/XP:

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

Note

Reinstall TCP/IP components if required!
If the "Internet Protocol TCP/IP" entry is missing, install the corresponding TCP/IP components and reboot your computer.
You will need the installation CD for Windows NT, 2000 or XP.

3. Then click on the **Properties...** button

4. The IP address, subnet mask and, where required, the client PC's gateway address appear in the Properties window. Note these values:

   Client PC IP address: ___ . ___ . ___ . ___
   Subnet mask: ___ . ___ . ___ . ___
   Gateway: ___ . ___ . ___ . ___

5. Now select the desired IP address for your fieldbus node.

Note

Assign the client PC a fixed IP address and note common subnet!
Note that the client PC, on which the BootP server is listed, must have a fixed IP address and that the fieldbus node and client PC must be in the same subnet.

6. Note the IP address you have selected:

   Fieldbus node IP address: ___ . ___ . ___ . ___

8.2.3.3 Assigning the IP address and Enable BootP

1. Based on the handling, which depends on the BootP program set, assign the required IP address for your fieldbus node.

2. Enable the query/response mechanism of the BootP protocol based on the handling, which depends on the BootP program set.

3. To apply the new IP address, use a hardware reset to restart your fieldbus node (interrupt the voltage supply for approx. 2 seconds).

8.2.3.4 Disabling BootP

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

You must then deactivate the BootP protocol so that the controller uses the configurated IP address from the EEPROM; this does away with the need for the BootP server to be permanently available.

Note

BootP must be disabled to assign the address permanently!
To apply the new IP address permanently in the fieldbus controller, BootP must be disabled.
This prevents the fieldbus coupler from receiving a new BootP request.

Note

The IP address is not lost when the BootP-Protocol is disabled!
If the BootP protocol is deactivated after addresses have been assigned, the stored IP address is retained, even after an extended loss of power, or when the controller is removed

You can disable in the Web-based Management System.

Disable BootP in the Web-based Management System

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

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

A dialog window then appears with a password prompt. This is provided for secure access and entails three different user groups: admin, guest and user.

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

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

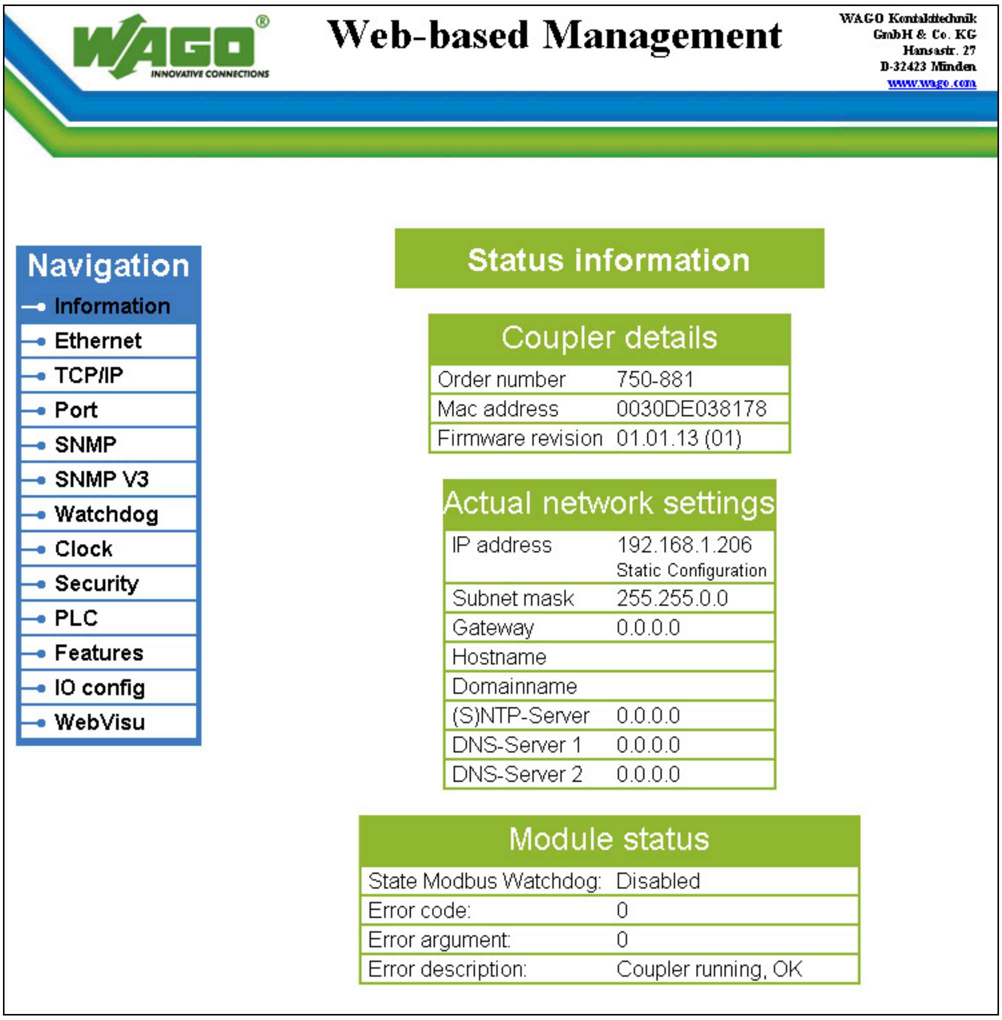


Figure 47: WBM page "Information"

Note

Disable the proxy server to display the web-based Management-System!
If these pages are not displayed for local access to the fieldbus nodes, you must define in the Web browser properties that, as an exception, no proxy server are to be used for the node IP address.

Note

The controller IP can be changed in the network by the DHCP server!
If DHCP is activated and an DHCP server is installed in the network (e.g. a router with DHCP server activated) addresses will be assigned automatically from the address range for the router after a loss of power (loss of 24 V DC power to controller). As a result, all controllers with activated DHCP will be assigned new IP addresses!

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

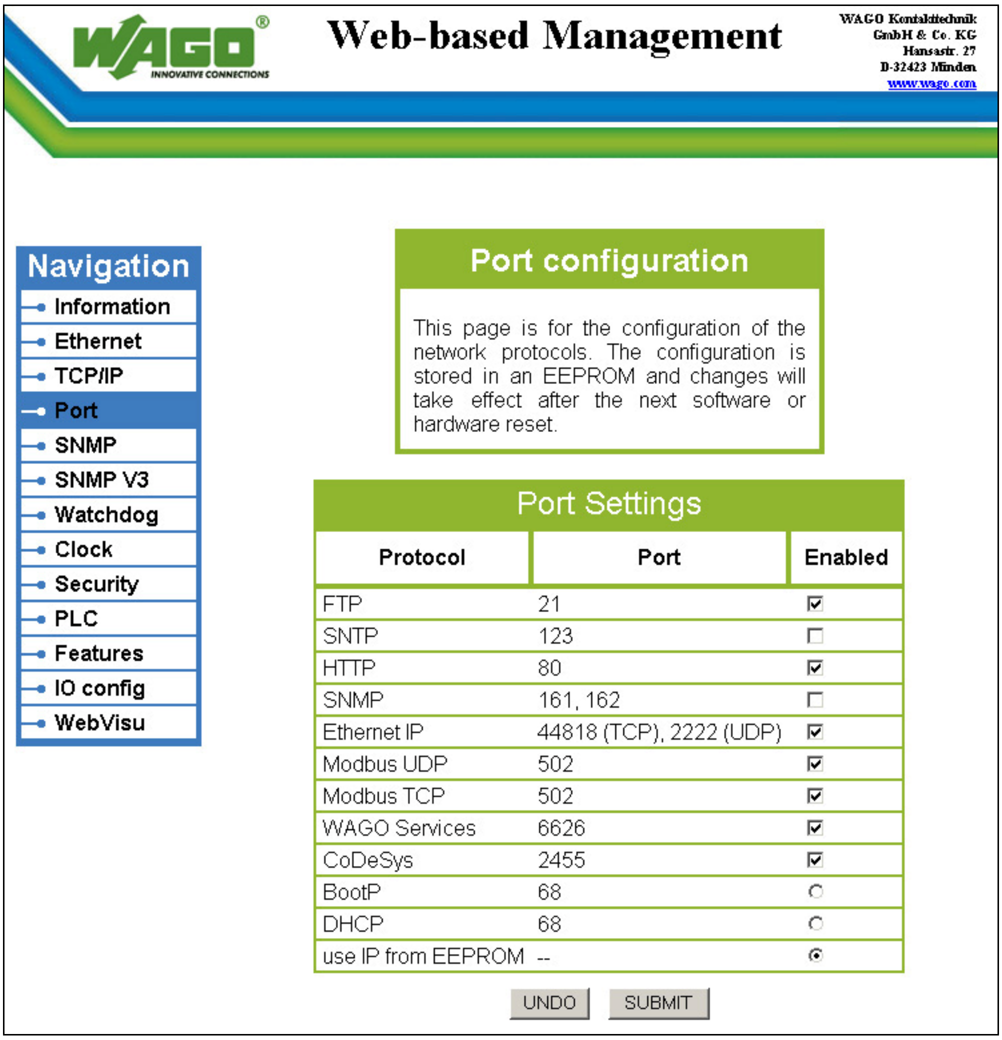


Figure 48: WBM page "Port"

You are shown a list of all the protocols supported by the controller.

5. Select the option "**DHCP**" or "**use IP from EEPROM**".
You have now deactivated the BootP protocol.

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

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

6. Click on **SUBMIT** and then switch off the power to the controller (hardware reset), or press down the mode selector switch. The protocol settings are then saved and the controller is ready for operation.

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

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

8.2.3.5 Reasons for Failed IP Address Assignment

- The controller MAC address does not correspond to the entry given in the "bootstrap.txt" file.
- The client on whom the BootP server is running is not located in the same subnet as the controller; i.e., the IP addresses do not match
  Example: Client IP: 192.168.0.10 and controller IP: 10.1.254.5
- Client and/or controller is/are not linked to the ETHERNET
- Poor signal quality (use switches or hubs)

8.3 Testing the Function of the Fieldbus Node

1. To ensure that the IP address is correct and to test communication with the fieldbus node, first turn off the operating voltage of the fieldbus node.

2. Create a non-serial connection between your client PC and the fieldbus node.

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

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

Information

More information about LED signaling
The exact description for evaluating the LED signal displayed is available in the section "Diagnostics", "LED Signaling".

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

4. In the DOS window, enter the command: "**ping** " followed by the IP address of your coupler in the following format:

   ping [space] XXX . XXX . XXX . XXX (=IP address)

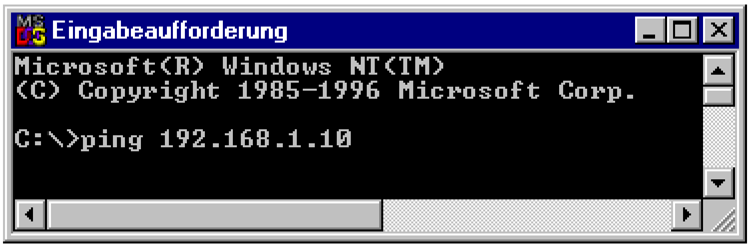


Figure 49: Example for the Function test of a Fieldbus Node

5. When the **[Enter]** key has been pressed, your PC will receive a query from the coupler, which will then be displayed in the DOS window.
   If the error message: "Timeout" appears, please compare your entries again to the allocated IP address and check all connections.

6. When the test has been performed successfully, you can close the DOS prompt.

The fieldbus node is now ready for communication.

8.4 Preparing the Flash File System

The flash file system must be prepared in order to use the Web interface of the fieldbus controller to make all configurations.

The flash file system is already prepared when delivered.
However, if the flash file system has not been initialized on your fieldbus controller or it has been destroyed due to an error, you must first extract it to the flash memory to access it.

NOTICE

Do not connect 750-920 Communication Cable when energized!
To prevent damage to the communications interface, do not connect or disconnect 750-920 Communication Cable when energized! The fieldbus coupler must be de-energized!

Note

Formatting erases data!
Note that formatting erases all data and configurations.
Only use this function when the flash file system has not been initialized yet or has been destroyed due to an error.

1. Switch off the supply voltage of the fieldbus controller.

2. Connect the communication cable 750-920 to the configuration interface of the fieldbus controller and to a vacant serial port on your computer.

3. Switch on the supply voltage of the fieldbus controller.

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

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

Information

More information about LED signaling
The exact description for evaluating the LED signal displayed is available in the section "Diagnostics", "LED Signaling".

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

5. In the top menu bar, select **Format** to format the file system.
   Formatting is complete when the status window displays "Formatting flash disk successfully done".

6. In the top menu bar, select **Extract** to extract the Web pages of the flash file system.
   This process takes a few seconds and is complete when the status window displays "Extracting files successfully done."

Note

Restart the Fieldbus coupler/controller after [Format]/[Extract]!
Make a restart of the fieldbus coupler/controller, so that the Web pages can be displayed after a Format/Extract.

8.5 Synchronizing the Real-Time Clock

The fieldbus controller's real-time clock enables a date and time indication for files in the flash file system.
At start-up, synchronize the real-time clock with the computer's current time.

There are two options to synchronize the real-time clock:

- Synchronize the real-time clock using **WAGO-ETHERNET-Settings**
- Synchronize the real-time clock using the **Web-based Management-System**

Synchronize the real-time clock using WAGO-ETHERNET-Settings

1. Switch off the supply voltage of the fieldbus controller.
2. Connect the communication cable 750-920 to the configuration interface of the fieldbus controller and to a vacant serial port on your computer.
3. Switch on the supply voltage of the fieldbus controller.

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

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

Information

More information about LED signaling
The exact description for evaluating the LED signal displayed is available in the section "Diagnostics", "LED Signaling".

4. Start the **WAGO Ethernet Settings** program.
5. Select the **Real-time Clock** tab.

Figure 50: Example of real-time clock synchronization in ETHERNET Settings

6. Click on the "Synchronize" button with the clock icon.

Synchronize the real-time clock using the Web-based Management-System

1. Launch a Web browser (e.g., MS Internet Explorer or Mozilla) and enter the IP address you have assigned your fieldbus node in the address bar.

2. Click **[Enter]** to confirm.
   The start page of the Web interface loads.

3. Select "Clock" in the left menu bar.

4. Enter your user name and password in the inquiry screen (default: user = "admin", password = "wago" or user = "user", password = "user").
   The HTML page "Clock configuration" loads:

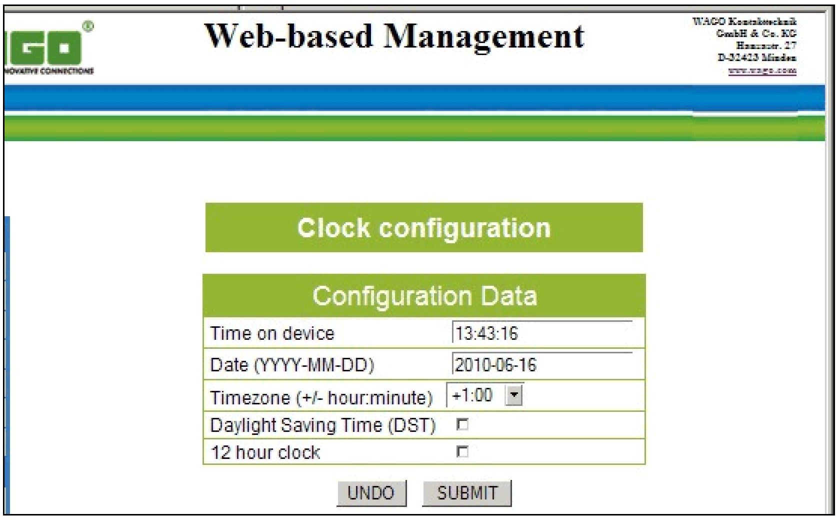


Figure 1: Example of WBM clock configuration

5. Set the values in the fields "Time on device", "Date" and "Timezone" to the current values and enable the "Daylight Saving Time (DST)" option if necessary.

6. Click on **[SUBMIT]** to apply the changes in your fieldbus node.

7. Restart the fieldbus node to apply the settings of the Web interface.

8.6 Restoring Factory Settings

To restore the factory settings, proceed as follows:

1. Switch off the supply voltage of the fieldbus controller.

2. Connect the communication cable 750-920 to the configuration interface of the fieldbus controller and to a vacant serial port on your computer.

3. Switch on the supply voltage of the fieldbus controller.

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

5. In the top menu bar, select **Default** and click **[Yes]** to confirm.

A restart of the fieldbus node is implemented automatically. The start takes place with the default settings.

9 Programming the PFC using WAGO-I/O-*PRO*

Using IEC 61131-3 programming, the Programmable Fieldbus Controller ETHERNET 750-881 can also utilize the function of a PLC in addition to the functions of a fieldbus coupler. Creation of an application program in line with IEC 61131-3 is performed using the programming tool WAGO-I/O-*PRO*.

Note

Activate option "CoDeSys" in the web-based Management System for programming!
Pay attention, the IEC 61131-3 programming of the controller via ETHERNET requires that the check box **CoDeSys** be activated at the Website "Port Configuration" (default).
You can, however, also connect the client PC and controller serially for programming using a programming cable.

A description of programming using WAGO-I/O-*PRO* is not included in this manual. The following sections, on the other hand, contain important information about creating projects in WAGO-I/O-*PRO* and about special modules that you can use explicitly for programming of the Programmable Fieldbus Controller ETHERNET.

Explanations are also provided as to how the IEC 61131-3 program is transferred and how suitable communication drivers are loaded.

Information

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

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

2. Under **File** / **New** create a new project

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

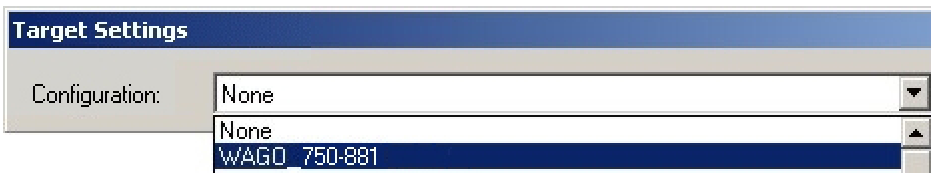


Figure 51: Dialog window for target system settings

3. Select the WAGO 750-881 ETHERNET Controller10/100 Mbit/s; digital and analog Signals750-881 by entering **WAGO_750-881** and then click **OK**.

4. In the dialog window that appears select the program type (AWL, KOP, FUP, AS, ST or CFC).

To ensure that you can access all I/O module data properly in your new project, first compile the I/O module configuration based on the existing fieldbus node hardware and map it in the configuration file "EA-config.xml".

This file defines whether write access is permitted to the modules from the IEC-61131-3 program, from the MODBUS/TCP or from Ethernet/IP.

As described below, this file can be generated via configuration using the WAGO I/O Configurator.

9.1 Configuration using the WAGO-I/O-PRO I/O Configurator

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

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

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

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

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

5. By right clicking on the entry **K Bus** and the command **Attach subelement** in the menu, you can select the required I/O module from the I/O module catalog.
   (In the new versions of the I/O Configurator open the I/O module catalog by additional clicking on the button **Add.**)

6. Attach it to the end of the **K Bus** structure using **Insert** and then clicking **OK**. In this case, the command "Insert element" is deactivated.

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

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

8. Position all of the required I/O modules until this arrangement corresponds to the configuration of the physical node.

Complete the tree structure in this process for each module in your hardware that sends or receives data.

Note

The terminal bus structure in the WAGO I/O Configurator must match the physical node structure!
The number of modules that send or receive data must correspond to the existing hardware (except for supply modules, copying modules or end modules, for example).
For the number of input and output bits or bytes of the individual I/O modules please, refer to the corresponding description of the I/O modules.

Information

Additional Information
To obtain further information about an I/O module, either select that module from the catalog, or in the current configuration and then click the button Data Sheet. The module is then shown in a separate window with its associated data sheet.
For the current version of the data sheets go to http://www.wago.com under Documentation.

9. Click **OK** to accept the node configuration and close the dialog window.

The addresses for the control system configuration are then recalculated and the tree structure for the configuration updated.

If required, you can also modify the authorization privileges for individual I/O modules if they are to be accessed via fieldbus (MODBUS/TCP/IP or Ethernet/IP). Initially, write access from the PLC is defined for each I/O module that is added. Proceed as follows to change this setting:

10. Click on a module in the configuration.

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

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

- PLC (standard setting) - Access from PLC
- fieldbus 1 - Access from MODBUS/TCP
- fieldbus 2 - Access from Ethernet/IP

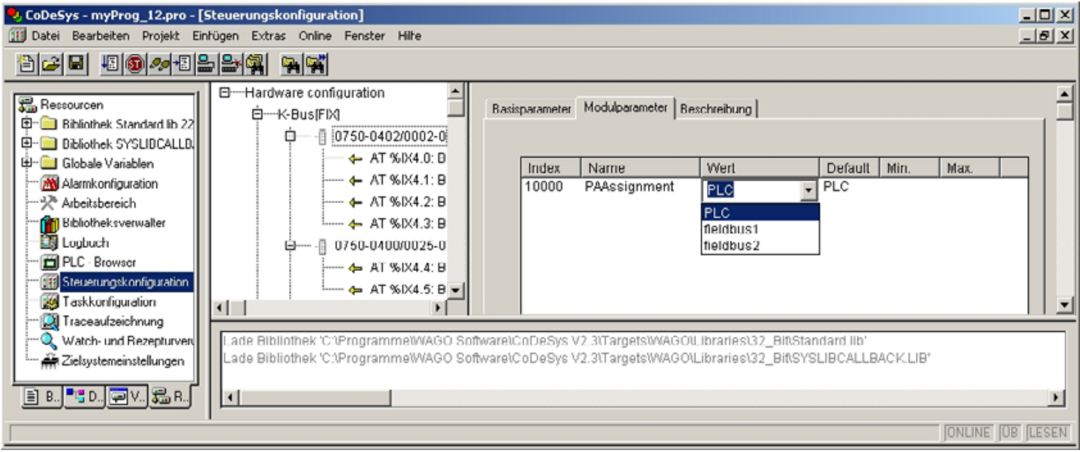


Figure 52: Write access via module parameters

After completing these settings you can begin the IEC-61131-3 programming.

An "EA-config.xml" configuration file is automatically generated and stored in the fieldbus controller, when you transfer the project (Menu **project > transfer/transfer all**) and download it in the fieldbus controller.

Note

Set "fieldbus1", when directly writing to a hardware address via MODBUS!
Set fieldbus 1 if you wish to write directly to a hardware address via MODBUS. Otherwise the modules will be allocated to the PLC, making writing from a different location impossible.

Information

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

9.1.1 Configuration using the "EA-config.xml" File

You can also create the file "EA-config.xml" using an editor and store it in the controller directory "/etc" by means of FTP.

Configuration using the file "EA-config.xml" that is already stored in the controller is described in this section.

Note

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

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

   ftp://[IP address of controller], e.g. ftp://192.168.1.201

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

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

3. Copy this file to a local directory on your PC and open it in an editor installed on your PC (e.g., "WordPad").

The file already contains the following syntax:

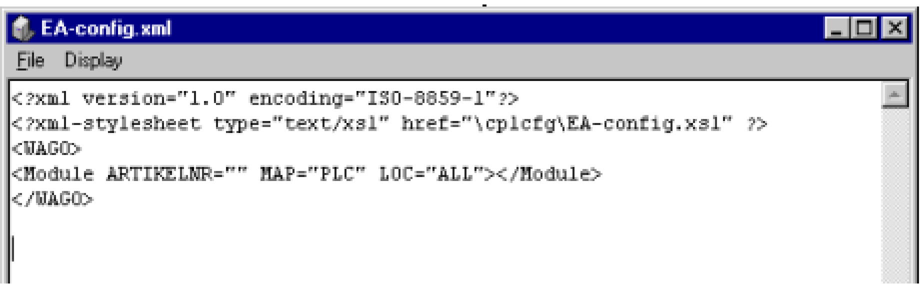


Figure 53: EA-config.xml

The fourth line contains the necessary information for the first I/O module.
The entry MAP=PLC assigns write access privileges to the IEC-61131-3 program for the first module.

4. If you want to enable access via MODBUS/TCP, replace "PLC" with "FB1" and for access from Ethernet/IP, replace "PLC" with "FB2":

   <Module ARTIKEL NUMBER=“ “ MAP=“**PLC**“ LOC=“ALL“>
   </Module>

<Module ARTIKEL NUMBER=“ “ MAP=“**FB1**“ LOC=“ALL“>
</Module>

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

Note

The number of line entries must correspond with the number of bus terminals used!
It is imperative that the number of line entries concurs with the number of existing hardware modules.

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

You can then begin with IEC-61131-3 programming.

Information

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

9.2 ETHERNET Libraries for WAGO-I/O-*PRO*

Various libraries are available in WAGO-I/O-*PRO* for different IEC 61131-3 programming tasks. These libraries contain function blocks that can be used universally to facilitate and accelerate the creation of programs.

Once the libraries have been integrated, function blocks, functions and data types will be available that you can use the same as ones you have specifically defined.

Information

Additional Information
All libraries are included on the installation CD for the software WAGO-I/O-*PRO* or in the Internet under: http://www.wago.com → Downloads → AUTOMATION

The libraries described below are specific to ETHERNET projects with WAGO-I/O-*PRO*.

Table 37: ETHERNET libraries for WAGO-I/O-*PRO*

Library	Description
Ethernet.lib	Function blocks for communication via ETHERNET
WAGOLibEthernet_01.lib	Function blocks that can set up a link to a remote server or client via TCP protocol to exchange data with any potential UDP server or client via UDP protocol
WAGOLibModbus_IP_01.lib	Function blocks that set up links with one or more slaves
ModbusEthernet_04.lib	Function blocks for data exchange with several MODBUS/TCP/UDP slaves Also a function block that provides a MODBUS server that maps the MODBUS services on a word array.
SysLibSockets.lib	Function block for access to sockets for communication via TCP/IP and UDP.
WagoLibSockets.lib	Function blocks for access to sockets for communication via TCP/IP and UDP Contains additional functions in addition to SysyLibSockets.lib.
Mail_02.lib	Function block for sending e-mails
WAGOLibMail_02.lib	Function block for sending e-mails
WagoLibSnmpEx_01.lib	Function blocks for sending SNMP-V1 traps together with the parameters for the type DWORD and STRING(120) (starting with software version SW >= 07).
WagoLibSntp.lib	Function blocks for setting and using the simple network time protocol (SNTP)
WagoLibFtp.lib	Function blocks for setting and using the file transfer protocol (FTP)
WAGOLibTerminalDiag.lib	Function blocks for the output of module, channel and diagnostic data of I/O modules that provide diagnostic data

Information

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

9.3 Functional Restrictions and Limits

The basis of WAGO-I/O-*PRO*, the standard programming system CoDeSys by 3S, has an integrated visualization. Dependend on the target, this visualization can be used in the variants "HMI", "TargetVisu" and "WebVisu".

The fieldbus controller supports the process variants "HMI" and "WebVisu". Depending on the version, there are technological limitations.

Several options for complex visualization objects "Alarm" and "Trend" are only provided by the "HMI" version. This applies, for example, to sending emails as a response to an alarm or for navigating through and generating historical trend data.

Compared with "HMI," the "WebVisu" on the fieldbus controller is executed within considerably tighter physical limits. Whereas the "HMI" can call upon the resources of a PC, the "WebVisu" operate within the following restrictions:

File system (2 MB):
The overall size of the PLC program, visualization files, bitmaps, log files, configuration files, etc. must fit into the file system.
The PLC browser delivers the amount of free disk space in response to the command "fds" (FreeDiscSpace).

Process data buffer (16 kB):
The WebVisu uses its own protocol for exchanging process data between applet and control system. In doing so, the process data is transmitted with ASCII coding. The pipe character ("|") separates two process values. For this reason, the required space of a process data variable in the process data buffer not only depends on the data type, but also on the process values itself. A "WORD" variable therefore occupies between one byte for the values 0...9 and five bytes for values greater than 10000. The selected format allows only a rough estimate of the space required for the individual process data in the process data buffer. If the size is exceeded, the WebVisu no longer works as expected.

The number of modules (1023/default):
The total size of the PLC program is determined, among other things, by the maximum number of modules. This value can be configured in the target system settings.

Computing power/processor time:
The 750-881 is based on a real-time operating system with pre-emptive multitasking. High-priority processes such as the PLC program will eliminate low-priority processes.
The web server supplies process data and applets for the web visualization.
Make sure when configuring tasks, that there is sufficient processor time available for all processes. The "freewheeling" task call option is not suitable in conjunction with the "WebVisu"; as in this case, the high-priority PLC program suppresses the web server. Instead of this, use the "cyclic" task call option with a realistic value.
The PLC browser provides an overview of the real execution times for all

CoDeSys tasks with the command "tsk".
If in a PLC program, operating system functions are used; e.g., for the handling of "sockets" or the "file system," these execution times are not taken into consideration covered by the command "tsk".

CTU counter:
The CTU counter operates in a value range of 0 to 32767.

Network load:
The Programmable Fieldbus Controller ETHERNET has one CPU responsible both for running the PLC program and for handling network traffic.
Ethernet communication demands that every telegram received is processed, regardless of whether it is intended for the Programmable Fieldbus Controller ETHERNET or not.

A significant reduction of the network load can be achieved by configuring the bandwidth limit of the integrated switch module or by using external "switches" instead of "hubs".

However, broadcast telegrams can either only be checked by the sender or with configurable switches that have broadcast limiting. A network protocol analyzer/monitor such as www.ethereal.com provides an overview of current network loading.

Note

Do not use bandwidth limits to increase the operational safety!
The bandwidth limit that can be configured in the WBM under the "Ethernet" link is not suitable for increasing the operating reliability of the "WebVisu", as in this case telegrams are ignored or rejected.

Information

Additional Information
The definition of hard benchmark data is not possible (due to the reasons mentioned above). For planning support, please use the application notes published online for relevant projects featuring the capability of Web visualization. This information is located at http://www.wago.com.

Note

Note the maximum number of write cycles of the EEPROM!
Fieldbus couplers/controllers save some information such as IP addresses and IP parameters in the EEPROM to make it available after a restart. The memory cycles of an EEPROM are generally limited. Beyond a limit of approx. 1 million write cycles, memory can no longer be assured. A defective EEPROM only becomes apparent after a restart by software reset or power-on.
Due to a bad checksum, the fieldbus coupler/controller then always starts with the default parameters.

The following functions use the EEPROM:

- **WAGO-I/O-*PRO***
  - **WagoLibDaylightSaving** SetDaylightSavings
  - **EthernetLib** SetNetworkConfig SetVariables
- **MODBUS**
  - Register 0x1035 Time Offset
  - Register 0x100B Watchdog parameters
  - Register 0x1028 Network configuration
  - Register 0x1036 Daylight saving
  - Register 0x1037 Modbus response delay
  - Register 0x2035 PI parameter
  - Register 0x2043 Default configuration

- **Ethernet/IP**
  - Class 0xF5
  - Class 0xF6
  - Class 0x64

- **Parameter assignments**
  - **BootP** new parameters
  - **DHCP** new parameters
  - **WAGO MIB** write access

9.4 General Information about IEC Tasks

Please note the following information when programming your IEC tasks:

Note

Use different priorities for IEC tasks!
IEC tasks must have different priorities, as otherwise an error will occur during translating of the application.

An interruption of IEC tasks is possible through tasks of higher priority!
An ongoing task may be interrupted by tasks with higher priorities. Execution of the task that has been interrupted is resumed only when there are no other higher-priority tasks to be executed.

Distortion of variables in overlapping areas of the process image!
If several IEC tasks utilize input or output variables with the same, or overlapping addresses in the process image, the values for the input or output variables may change while the IEC task is being executed!

Observe waiting periods of free-running tasks!
Running tasks are halted after each task cycle for half the time that the task proper requires (min. 1 ms). Execution of the task is then resumed.

Example: 1st Task 4 ms → Waiting period 2 ms
2nd Task 2 ms → Waiting period 1 ms

The default task is created by default!
If no task has been defined in the task configuration, a running default task is created during translation. This task, called "Default task," is recognized by this name in the firmware, meaning that the name "Default task" can not be used for other task names.

Observe the watchdog sensitivity for cyclic tasks!
The watchdog sensitivity indicates how many times the watchdog time is exceeded for an even to be triggered. You set the sensitivity in WAGO-I/O-*PRO* under Register **Resources > Task Configuration** for Cyclical Tasks. The values 1 and 0 are equivalent with regard to sensitivity. A sensitivity value of 0 or 1 results in the watchdog event being triggered when the watchdog time is exceeded on time. With a sensitivity value of 2, for instance, the watchdog time must be exceeded in two consecutive task cycles in order for the watchdog event to be triggered.

The following applies to cyclic tasks with watchdog activated:

Note

Reference for Watchdog Settings!
For each tasks created, a watchdog can be enabled that monitors the execution time of a task.
If the task runtime exceeds the specified watchdog time (e.g., t#200 ms), then the watchdog event has occurred.
The runtime system stops the IEC program and reports an error.

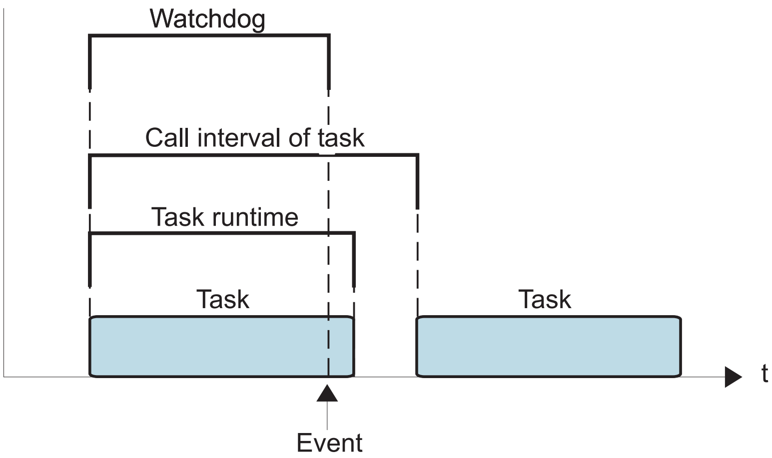


Figure 54: Watchdog runtime is less than the task runtime

If the watchdog time set is greater than the call interval of the task, then the watchdog is restarted for each task call interval.

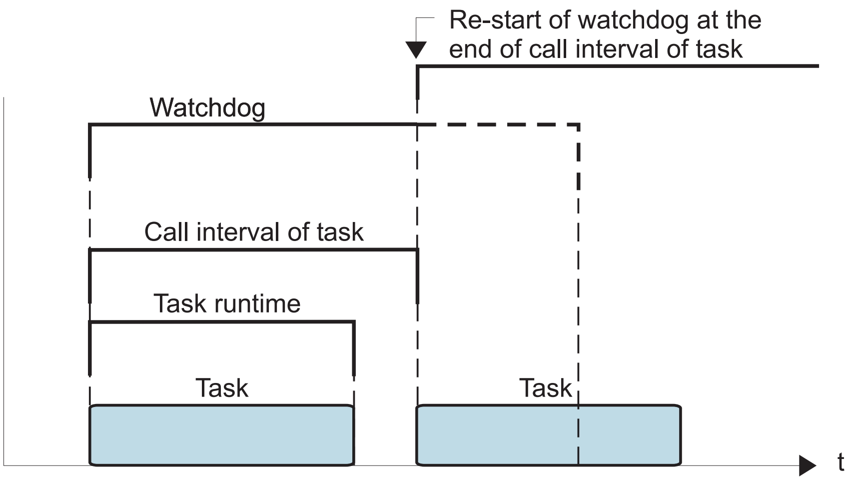


Figure 55: Watchdog runtime is greater than the task call interval

To cyclic tasks applies:

Note

Cyclic tasks with > 30 min. call intervals not possible!
Cyclic tasks with a call interval of more than 30 minutes are not possible.

9.4.1 IEC Task Sequence

1. Determine the system time (tStart).
2. If no full internal bus cycle has run since the last time the outputs were written:
   → Wait until the next internal bus cycle is completed.
3. Reading of inputs and reading back of the outputs from the process image.
4. If the application program has been started.
   → Execute the program codes for this task.
5. Writing of the outputs to the process image.
6. Determine the system time (tEnd).
   → tEnd - tStart = runtime for the IEC task

9.4.2 Overview of Most Important Task Priorities

Table 38: Task processing

Task	Importance of the execution
Internal bus task, fieldbus task	of priority before all others
Normal task	after the internal bus and fieldbus tasks
PLC-Comm task	after the normal tasks
Background task	after the PLC-Comm tasks

I/O Bus Task / Fieldbus Task (Internal)
The I/O Bus task is an internal task, which updates the I/O module data from the process image. Fieldbus tasks are triggered by fieldbus events (communications); therefore, they only use processing time when the fieldbus is active (MODBUS/Ethernet/IP).
Normal task (IEC tasks 1-10)
IEC tasks with this priority may be interrupted by the internal bus tasks. Therefore, configuration for the connected modules and communication via fieldbus with the watchdog activated for the task call interval must be taken into account here.

PLC-Comm task (internal)
The PLC-Comm task is active when logged in and takes up communication with the CoDeSys gateway.

Background task (IEC-Task priorities 11-31 that can be set in CoDeSys)
All internal tasks have a priority higher than that for the IEC background tasks. These tasks are therefore very well-suited for performing time-intensive and non-critical time tasks, such as calling up functions in the SysLibFile.lib.

Information

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

9.5 System Events

In place of a task, a system event can also call up a project module for processing.

The system events to be employed for this depend on the target system. These events consist of the list of supported standard system events for the control system and any other manufacturer-specific events which may have been added.

Possible events, for example: Stop, Start, Online change.

A complete list of all system events is provided at WAGO-I/O-*PRO* in tab **Resources > Task configuration > System events**.

9.5.1 Enabling/disabling system events

1. Open the register **resources > task configuration > system events** in WAGO-I/O-*PRO* (see the following Figure).

2. In order to call up a module via an event, activate the entries by setting a hatch mark in the respective control boxes.

3. Disable the control boxes by removing the hatch marks through a mouse click.

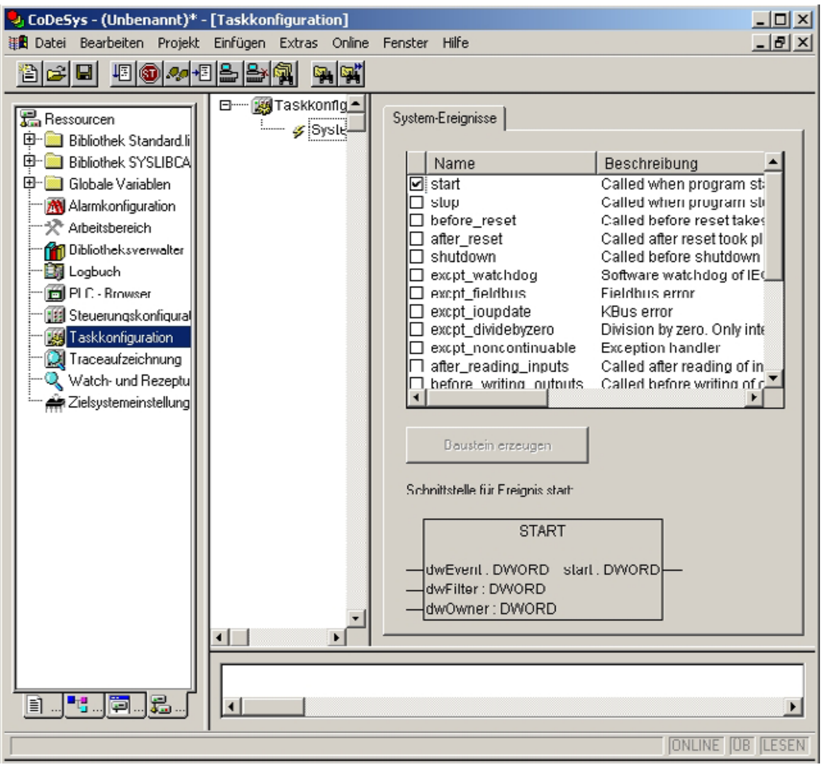


Figure 56: System events

Information

Additional Information:
Allocation of the system events to the specific modules to be called up is clarified in the manual for the programming tool WAGO-I/O-*PRO* in the Internet under: http://www.wago.com → Documentation → WAGO Software → WAGO-I/O-*PRO* → 759-333

9.6 Transfer the IEC program to the controller

Transfer from the PC to the controller of the program for the created IEC-61131-3 application can be performed two ways (see following sections).

- Direct transfer via serial RS-232 port
- Transfer by means of TCP/IP via fieldbus

Suitable communication drivers are required for transfer; these can be loaded and configured using WAGO-I/O-*PRO*.

Note

Check/adjust communications parameters of the driver
When selecting the desired driver, watch for the proper settings and adjustments of the communications parameters (see the following description).

Note

"Reset" and "Start" are required to set the physical outputs!
The initialization values for the physical outputs are not set immediately after downloading. Select **Online** > **Reset** and subsequently **Online** > **Start** in the menu bar of WAGO I/O-*PRO* to set the values.

Note

Stop application before generating large boot projects!
Stop the WAGO-I/O-*PRO* application via **Online** > **Stop** before generating a very large boot project, since this may otherwise cause stopping the internal bus. You can restart the application after creating the boot project.

Note

Handling persistent data affects the program start!
Depending on the variable type, the number and sizes of the persistent data and their combination, such as in function modules, handling with persistent data can delay the program start by an extended initialization phase.

Information

Additional Information
The following description is used for fast access. For details on installing missing communication drivers and using the software, refer to "WAGO-I/O-*PRO*" available in the Internet under:
www.wago.com → Service → Downloads → Documentation → WAGO Software → WAGO-I/O-*PRO*.

9.6.1 Transfer via Serial Service Port

Note

Watch the position of the mode selector switch when accessing the controller!
Prerequisite for the access to the fieldbus controller is that the operating mode switch of the controller, which is located behind the cover of the fieldbus controller next to the service interface, is in the center or top position.

Use the WAGO communication cable to set up a physical connection via serial service port. This cable is included in the scope of supply for the IEC-61131-3 programming tool (order no. 759-333), or can be procured as an accessory item under order no. 750-920.

NOTICE

Do not connect 750-920 Communication Cable when energized!
To prevent damage to the communications interface, do not connect or disconnect 750-920 Communication Cable when energized! The fieldbus controller must be de-energized!

1. Check that the controller mode selector switch is set to the center or top position.
   If this is not the case, move the mode selector switch to the center or top position.

2. Use the WAGO communication cable to connect a COM port of your PC to the controller communication port.

A communication driver is required for serial data transfer. This driver and its parameters must be entered in the WAGO-I/O-*PRO* in the dialog window "Communication parameters".

3. Start the WAGO-I/O-*PRO* software **under Start > Programs > WAGO Software > WAGO-I/O-*PRO*.**

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

The dialog window "Communication parameters" then appears. The channels of the currently connected gateway servers are shown on the left side of the dialogue and the already installed communications drivers are shown below. This window is empty in its default settings.

5. Click **New** to set up a link and then enter a name, such as RS-232 Connection.

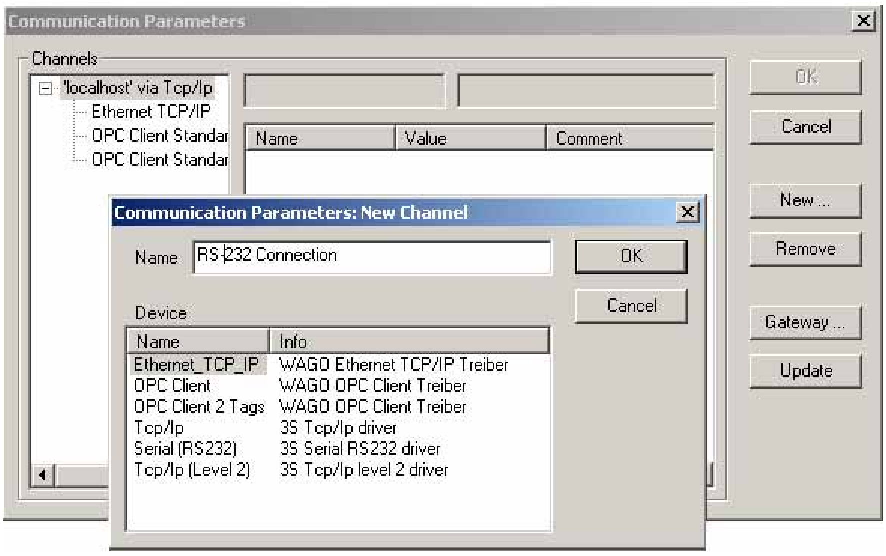


Figure 57: Dialog window "Communication parameters"

6. In the selection window, mark the required driver in the right side of the window, Serial (RS-232) 3S Serial RS-232 driver, to configure the serial link between the PC and the controller.

The following properties for the serial port are shown in the center dialog window:

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

7. If necessary, change the entries according to the above values by clicking on the respective value and editing it.

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

The RS-232 port is now configured for transferring the application.

9. Under **Online**, click the menu item **Login** to log in to the controller

The WAGO-I/O-*PRO* Server is active during online operation. The communication parameters can not be called up during this time.

Depending on whether a program is already present in the controller, a window will appear asking whether a (new) program should be loaded.

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

11. In menu **Online,** click on **Create Boot project.**

You compiled project will also be executed by this method, if you restart the controller or if there is a power failure.

12. Once the program has been loaded, start program processing in the menu **Online**, menu item **Start**.

This command starts the processing of your program in the control system or in the simulation.

"ONLINE" and "RUNNING" will then appear at the right of the status bar.

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

9.6.2 Transfer via Fieldbus and ETHERNET

The physical link between the PC and the controller is set up via fieldbus. An appropriate communication driver is required for data transfer. The driver and its parameters must be entered in the WAGO-I/O-*PRO* in the dialog window "Communication parameters".

Note

Controller needs IP address for access!
The controller must have an IP address before it can be accessed. The operating mode switch, which is located behind the cover of the fieldbus controller next to the service interface, must be in the center or top position.

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

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

The dialog window "Communication parameters" then appears. The channels of the currently connected gateway servers are shown on the left side of the dialogue and the already installed communications drivers are shown below. This window is empty in its default settings.

3. Click **New** to set up a connection and then specify a name, e.g. TcpIp connection.

4. Mark the required TCP/IP driver in the right side of the dialog window to configure the link between the PC and the controller via ETHERNET.
   Use the new driver version "Tcp/Ip" (3S Tcp/Ip driver).

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

- IP address: IP address of your controller
- Port number: 2455
- Motorolabyteorder: No
- Debug level: 16#0000

5. Change any entries as you may require.

6. Confirm with **OK**.

You have now configured the TCP/IP link with the communication parameters/drivers.

7. Under **Online**, click the menu item **Login** to log in to the controller

The WAGO-I/O-*PRO* Server is active during online operation. The communication parameters can not be called up during this time.

Depending on whether a program is already present in the controller, a window will appear asking whether a (new) program should be loaded.

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

9. In menu **Online,** click on **Create Boot project.**

You compiled project will also be executed by this method, if you restart the controller or if there is a power failure.

10. Once the program has been loaded, start program processing in the menu **Online**, menu item **Start**.

This command starts the processing of your program in the control system or in the simulation.

"ONLINE" and "RUNNING" will then appear at the right of the status bar.

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

10 Configuring via the Web-Based Management System (WBM)

An internal file system and an integrated Web server can be used for configuration and administration of the system. Together, they are referred to as the Web-Based Management System (WBM).

The HTML pages saved internally provide you with information about the configuration and status of the fieldbus node. In addition, you can also change the configuration of the device here.
You can also save HTML pages created yourself via the implemented file system.

Note

Always restart after making changes to the configuration!
The system must always be restarted for the changed configuration settings to take effect.

1. To open the WBM, launch a Web browser (e.g., Microsoft Internet Explorer or Mozilla Firefox).

2. Enter the IP address of the fieldbus coupler/controller in the address bar (192.168.1.1 by default or as previously configured).

3. Click **[Enter]** to confirm.
   The start page of WBM loads.

4. Select the link to the desired HTML page in the left navigation bar.
   A query dialog appears.

5. Enter your user name and password in the query dialog (default: user = "admin", password = "wago" or user = "user", password = "user").
   The corresponding HTML page is loaded.

6. Make the desired settings.

7. Press **[SUBMIT]** to confirm your changes or press **[UNDO]** to discard the changes.

8. Restart the system to apply the settings.

10.1 Information

The WBM page "Information" contains an overview of all important information about your fieldbus coupler/controller.

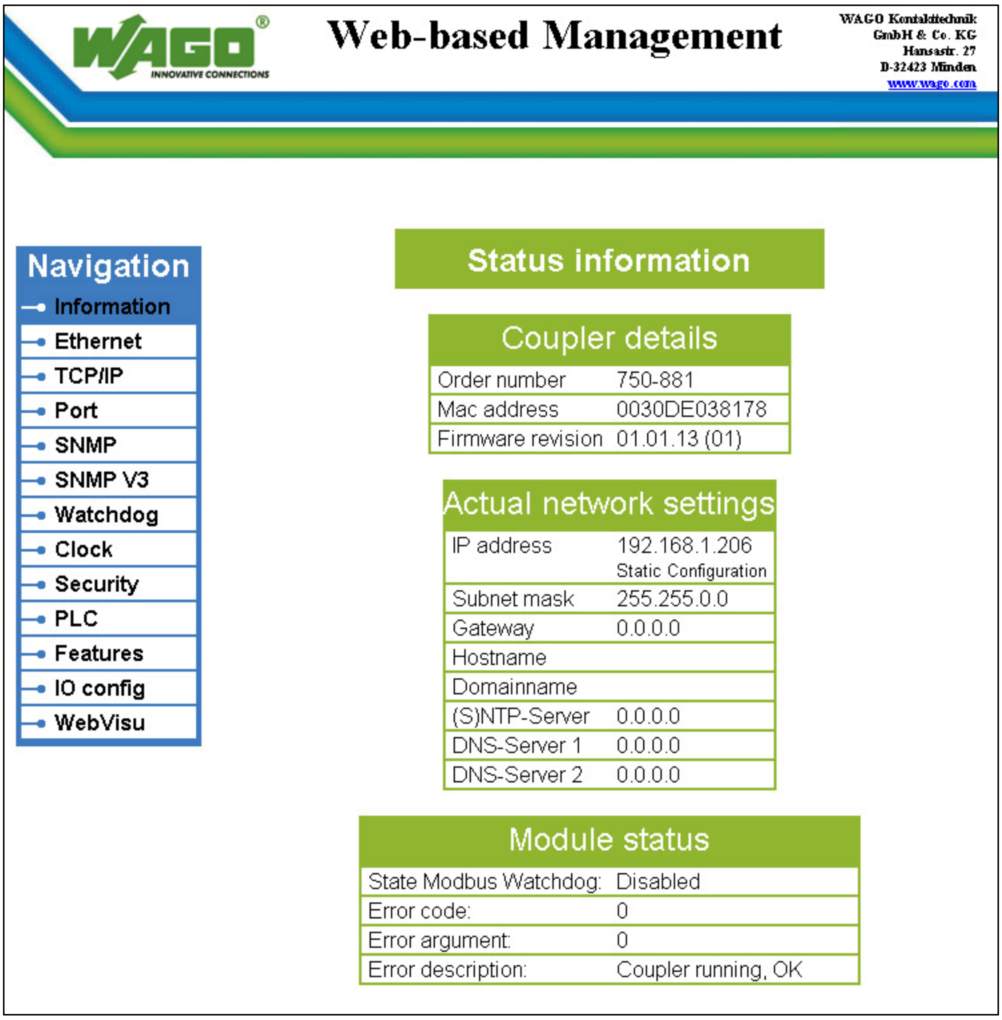


Figure 58: WBM page "Information"

Table 39: WBM page "Information"

Coupler details			
Entry	**Default**	**Value (example)**	**Description**
Order number	750-881/000-000	750-881/000-000	Order number
Mac address	0030DEXXXXXX	0030DE000006	Hardware MAC address
Firmware revision	kk.ff.bb (rr)	01.01.09 (00)	Firmware revision number (kk = compatibility, ff = functionality, bb = bugfix, rr = revision)
Actual network settings			
Entry	**Default**	**Value (example)**	**Description**
IP address	192.168.1.1	192.168.1.80	IP address, Type of IP address assignment
Subnet mask	255.255.255.0	255.255.255.240	Subnet mask
Gateway	0.0.0.0	192.168.1.251	Gateway
Hostname	___	___	Host name (not assigned here)
Domainname	___	___	Domain name (not assigned here)
(S)NTP server	0.0.0.0	0.0.0.0	Address of (S)NTP server
DNS server 1	0.0.0.0	0.0.0.0	Address of first DNS server
DNS server 2	0.0.0.0	0.0.0.0	Address of second DNS server
Module status			
Entry	**Default**	**Value (example)**	**Description**
State Modbus Watchdog	Disabled	Disabled	Status of Modbus Watchdog
Error code	0	10	Error code
Error argument	0	5	Error argument
Error description	Coupler running, OK	Mismatch in CoDeSys IO-configuration	Error description

10.2 Ethernet

Use the "Ethernet" HTML page to set the data transfer rate and bandwidth limit for each of the two switch ports for data transfer via Ethernet.

Figure 59: WBM-Seite „Ethernet“

Table 40: WBM page "Ethernet"

Phy Configuration				
Entry	**Default**			**Description**
Enable Port	☑			☑ Enable Port 1/Port 2
				☐ Disable Port 1/Port 2
Enable autonegotiation	⊙			⊙ Enable Autonegotiation Automatically set the best possible transmission speed with "Enable Autonegotiation".
				○ Enable Autonegotiation
10 MBit Half Duplex 10 MBit Full Duplex 100 MBit Half Duplex 100 MBit Full Duplex	○ ○ ○ ○			Select half or full duplex for the ETHERNET to configure a fixed transmission speed 10 or 100 MBit
Chip Configurtaion				
Entry	**Default value**			**Description**
Fast Aging	☐			☑ Enable "Fast Aging" "Fast Aging" ensures that the cache for the MAC addresses is cleared faster in the switch. This may be required if a redundancy system (e.g., using a Jet-Ring network or comparable technology) needs to be set up.
				☐ Disable "Fast Aging"
Misc. Configuration				
Entry	**Port**			**Description**
	1	**2**	**in-ternal**	
Input Limit Rate	No Limit ▾			The Input Limit Rate limits network traffic when receiving. The rate is indicated in megabytes or kilobytes per second. If the limit is exceeded, packets are lost.
Output Limit Rate	No Limit ▾			The Output Limit Rate limits network traffic when sending. The rate is indicated in megabytes or kilobytes per second. If the limit is exceeded, packets are lost.
BC protection	☐	☐		☑ Broadcast Protection limits the number of broadcast telegrams per unit of time. If protection is on, the broadcast packets are limited at 100 Mbit to 8 packets per 10 ms and at 10 Mbit to 8 packets per 100 ms. If the limit is exceeded, packets are lost.
				☐ Broadcast Protection disabled.
Port Mirror	☐			☑ Enable port mirroring Port Mirroring is used for network diagnostics. Packets are mirrored from one port (mirror port) to another (sniffer port).
				☐ Disable port mirroring
Sniffer Port	○	⊙		Select the sniffer port the mirror port should be mirrored to.
Mirror Port	⊙	○	○	Select the mirror port which should be mirrored to the sniffer port.
Ethernet MTU	1500			Maximum packet size of a protocol, which can be transferred without fragmentation ("Maximum Transmission Unit" - MTU)

Note

Set the MTU value for fragmentation only!
Only set the value for MTU, i.e., the maximum packet size between client and server, if you are using a tunnel protocol (e.g., VPN) for ETHERNET communication and the packets must be fragmented.
Setting the value is independent of the transmission mode selected.

Note

Configure ETHERNET transmission mode correctly!
A fault configuration of the ETHERNET transmission mode may result in a lost connection, poor network performance or faulty performance of the fieldbus coupler/controller.

Note

All ETHERNET ports cannot be disabled!
Both ETHERNET ports can be switched off. If both ports are disabled and you press **[SUBMIT]**, the selection is not applied and the previous values are restored.

10.3 TCP/IP

You can configure network addressing and network identification on the "TCP/IP" HTML page.

Note

Set the DIP switch to "0" and enable "use IP from EEPROM"!
Before you change parameters on this page, set the DIP switch to zero and on the "Port configuration" WBM page, set the "use IP from EEPROM" option!
If these conditions are not met, the DIP switch settings are applied instead.

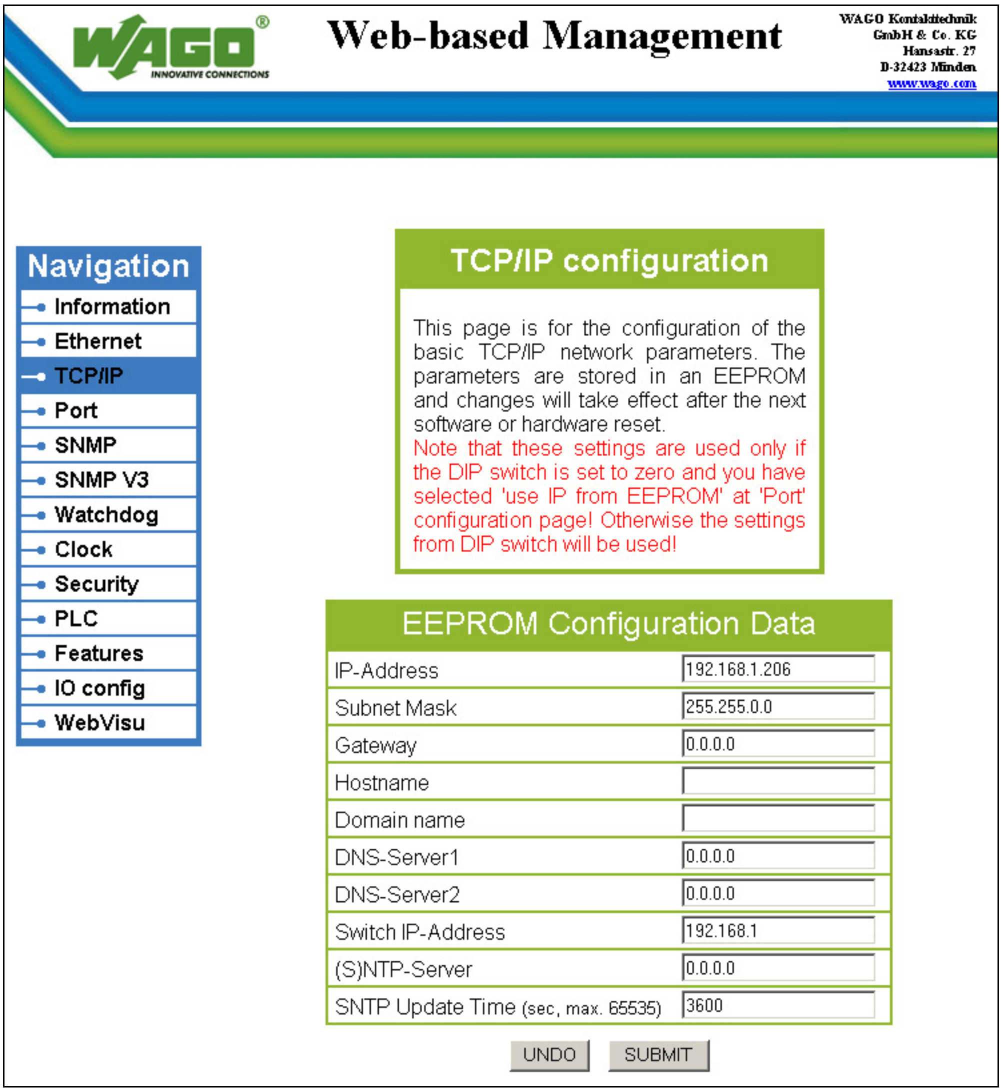


Figure 60: WBM page "TCP/IP"

Table 41: WBM page „TCP/IP“

Configuration Data			
Entry	**Default**	**Value (example)**	**Description**
IP address	192.168.1.0	192.168.1.200	Enter IP address
Subnet mask	255.255.255.0	255.255.255.0	Enter subnet mask
Gateway	0.0.0.0	0.0.0.0	Enter gateway
Host name			Enter host name
Domain name			Enter domain name
DNS Server1	0.0.0.0	0.0.0.0	Enter IP address of the first DNS server
DNS Server2	0.0.0.0	0.0.0.0	Enter optional IP address of the second DNS server
Switch IP-Address	192.168.1	192.168.5	Network address for the configuration of the IP address with DIP switch
(S)NTP Server	0.0.0.0	0.0.0.0	Enter IP address of the (S)NTP server
SNTP Update Time (sec. max. 65535)	0	0	Enter the delay after which the (S)NTP server requests the network time again

10.4 Port

Use the "Port" HTML page to enable or disable services available via the IP protocol.

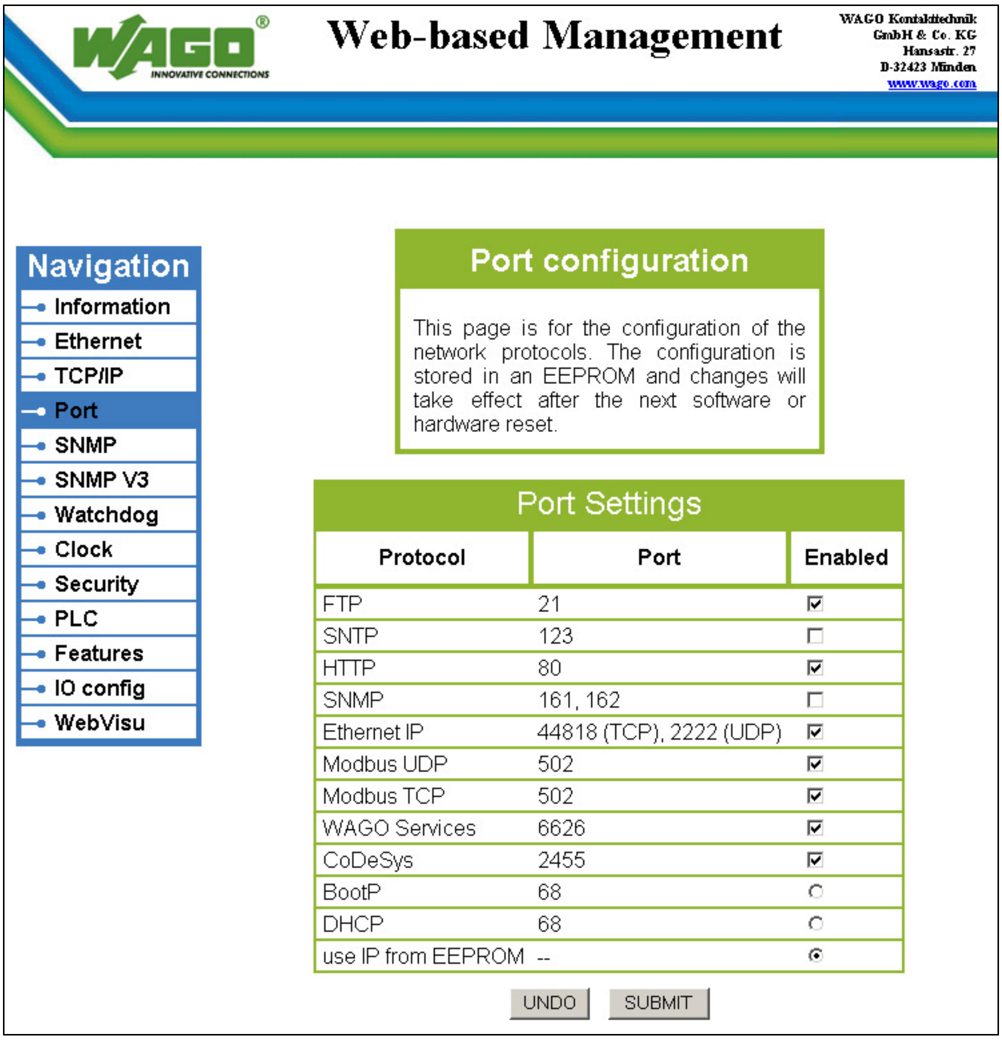


Figure 61: WBM page "Port"

Table 42: WBM page "Port"

Port Settings			
Entry	**Entry**	**Entry**	
FTP (Port 21)	Enabled ☑	☑	activating "File Transfer Protocol"
		☐	deactivating "File Transfer Protocol"
SNTP (Port 123)	Enabled ☐	☑	activating "Simple Network Time Protocol"
		☐	deactivating "Simple Network Time Protocol"
HTTP (Port 80)	Enabled ☑	☑	activating "Hypertext Transfer Protocol"
		☐	deactivating "Hypertext Transfer Protocol"
SNMP (Port 161, 162)	Enabled ☐	☑	activating "Simple Network Management Protocol"
		☐	deactivating "Simple Network Management Protocol"
Ethernet IP (TCP-Port 44818, UDP-Port 2222)	Enabled ☐	☑	activating ETHERNET/IP protocol
		☐	deactivating ETHERNET/IP protocol
Modbus UDP (Port 502)	Enabled ☑	☑	activating MODBUS/UDP protocol
		☐	deactivating MODBUS/UDP protocol
Modbus TCP (Port 502)	Enabled ☑	☑	activating MODBUS/TCP protocol
		☐	deactivating MODBUS/TCP protocol
WAGO Services (Port 6626)	Enabled ☑	☑	activating WAGO services
		☐	de activating WAGO services
CoDeSys (Port 2455)	Enabled ☑	☑	activating WAGO-I/O-*PRO*
		☐	deactivating WAGO-I/O-*PRO*
BootP (Port 68)	Enabled ◉	◉	activating "Boots Trap Protocol"
		○	deactivating "Boots Trap Protocol"
DHCP (Port 68)	Enabled ○	◉	activating "Dynamic Host Configuration Protocol"
		○	deactivating "Dynamic Host Configuration Protocol"
use IP from EEPROM	Enabled ○	◉	activating use of IP address from EEPROM
		○	deactivating use of IP address from EEPROM

Note

Alternative IP address assignment!
You can only select the DHCP, BootP and "use IP from EEPROM" settings as an alternative!

10.5 SNMP

On the HTML page "SNMP", you can perform the settings for the Simple Network Management Protocol.

SNMP is a standard for device management within a TCP/IP network. The Simple Network Management Protocol (SNMP) is responsible for transporting the control data that allows the exchange of management information, the status and statistic data between individual network components and a management system.

The fieldbus coupler/controller supports SNMP in versions 1, 2c and 3.

The SNMP of the ETHERNET TCP/IP controller includes the general MIB according to RFC1213 (MIB II).

SNMP is processed via port 161. The port number for SNMP traps (agent messages) is 162.

Note

Enable port 161 and 162 to use SNMP!
Enable ports 161 and 162 in the WBM in menu "port", so that the fieldbus coupler/controller can be reached via SNMP. The port numbers cannot be modified.

Note

Modify parameter via WBM or SNMP objects!
However, parameters that can be set on the html pages can also be changed directly by the appropriate SNMP objects.

Information

Additional Information:
Additional information for SNMP, the Management Information Base (MIB) and traps (event messages via SNMP) may be obtained from chapter "Fieldbus communications" > "Communications protocols" > "SNMP (Simple Network Management Protocol)".

Note that the settings for SNMPV1/V2c and SNMPV3 are separate from each other: The different SNMP versions can be activated or used in parallel or individually on a fieldbus controller.

10.5.1 SNMP V1/V2c

The SNMP version 1/2c represents a community message exchange. The community name of the network community must thereby be specified.

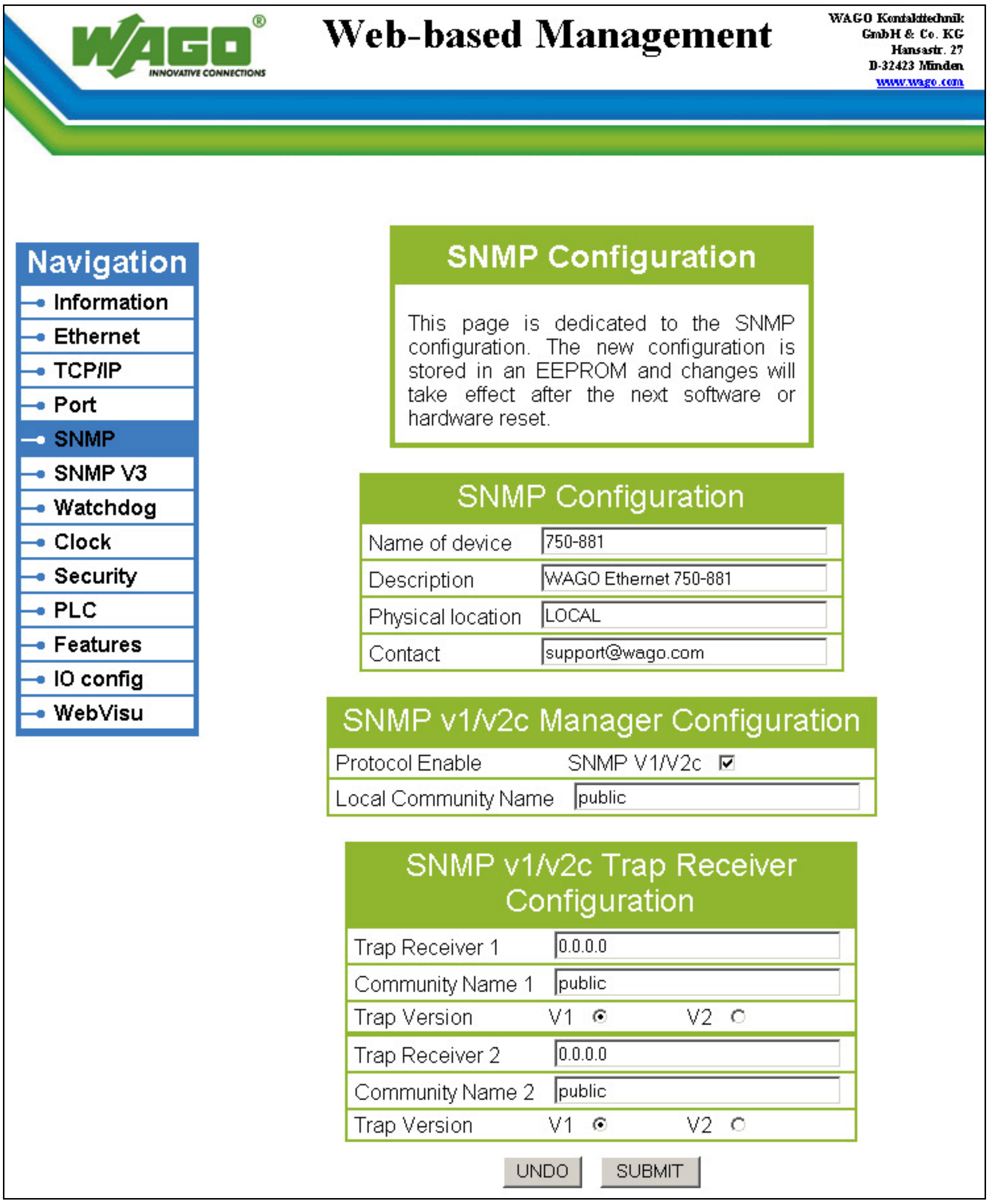


Figure 62: WBM page "SNMP"

Table 43: WBM page "SNMP"

SNMP Configuration		
Entry	**Value (Default)**	**Description**
Name of device	750-881	Device name (sysName)
Description	Programmable Fieldbus Controller ETHERNET 750-881	Device description (sysDescription)
Physical location	LOCAL	Location of device (sysLocation)
Contact	support@wago.com	E-mail contact address (sysContact)

SNMP v1/v2 Manager Configuration		
Entry	**Value (Default)**	**Description**
Protocol Enable	SNMP V1/V2c ☑	☑ Activating SNMP Version 1/2c
		☐ Deactivating SNMP-Version 1/2c
Local Community Name	public	Used community name
SNMP v1/v2 Trap Receiver Configuration		
Entry	**Value (Default)**	**Description**
Trap Receiver 1	0.0.0.0	IP address of **1.** used SNMP manager
Community Name 1	public	**1.** Community name of the network community used
Trap Version	V1 ◉ V2 ○	V1 ◉ V2 ○ Activating Traps Version 1
		V1 ○ V2 ◉ Activating Traps Version 2
Trap Receiver 2	0.0.0.0	IP address of **2.** used SNMP manager
Community Name 2	public	**2.** Community name of the network community used
Trap Version	V1 ◉ V2 ○	V1 ◉ V2 ○ Activating Traps Version 1
		V1 ○ V2 ◉ Activating Traps Version 2

10.5.2 SNMP V3

In SNMP version 3, exchanging messages is user-related. Each device, that knows the passwords set via WBM, may read or write values from the controller. In SNMPv3, user data from SNMP messages can also be transmitted in encoded form. This is why SNMPv3 is often used in safetyrelated networks.

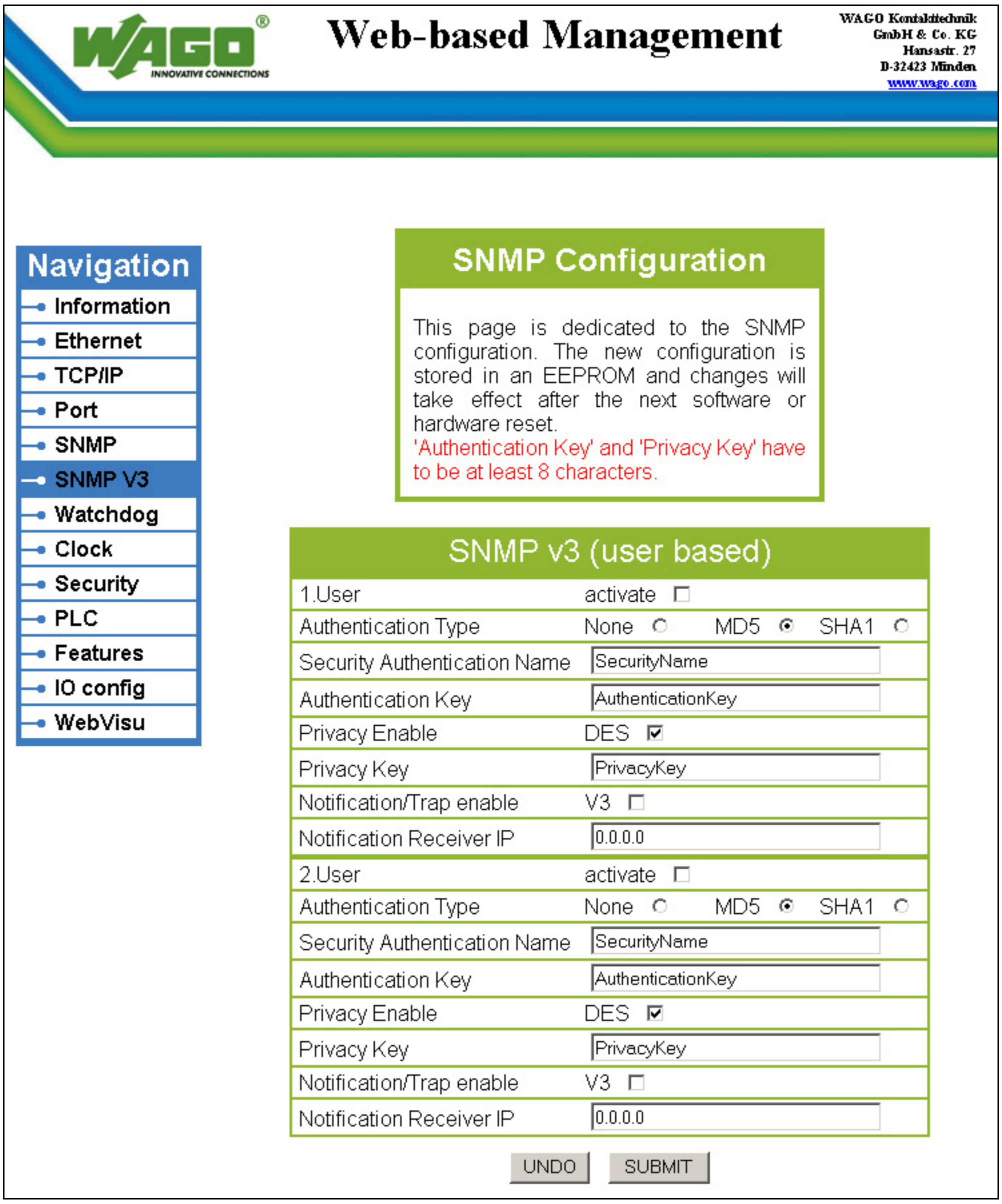


Figure 63: WBM page "SNMP V3"

SNMP v3 (user based)		
Entry	**Value (Example)**	**Description**
1. User / 2. User	activate ☑	☑ Activating user 1 or 2 ☐ Deactivating user 1 or 2
Authentification Type	None ○ MD5 ◉ SHA1 ○	None ◉ MD5 ○ SHA1 ○ No encryption of the authentication None ○ MD5 ◉ SHA1 ○ Encryption of the authentication with MD5 None ○ MD5 ○ SHA1 ◉ Encryption of the authentication with SHA1
Security Authentification Name	Security Name	Enter the name, if the "authentification type“ MD5 or SHA1 has been selected
Authentification Key	Authentification Key	Enter the password with at least 8 characters, if "authentification type“ MD5 or SHA1 has been selected
Privacy Enable	DES ☑	☑ Activate the DES encryption of the data ☐ Deactivate the DES encryption of the data
Privacy Key	Privacy Key	Enter the password of at least 8 characters in the encryption with DES
Notification/ Trap enable	V3 ☑	☑ Activate the notification traps of the SNMP version 3 ☐ Deactivate the notification traps of the SNMP version 3
Notification Receiver IP	192.168.1.10	IP address of the notification manager

Two independent SNMPv3 users can be defined and activated via the html page (user 1 and user 2).

10.6 Watchdog

Click the link "Watchdog" to go to a Web site where you can specify the settings for the connection and MODBUS watchdog.

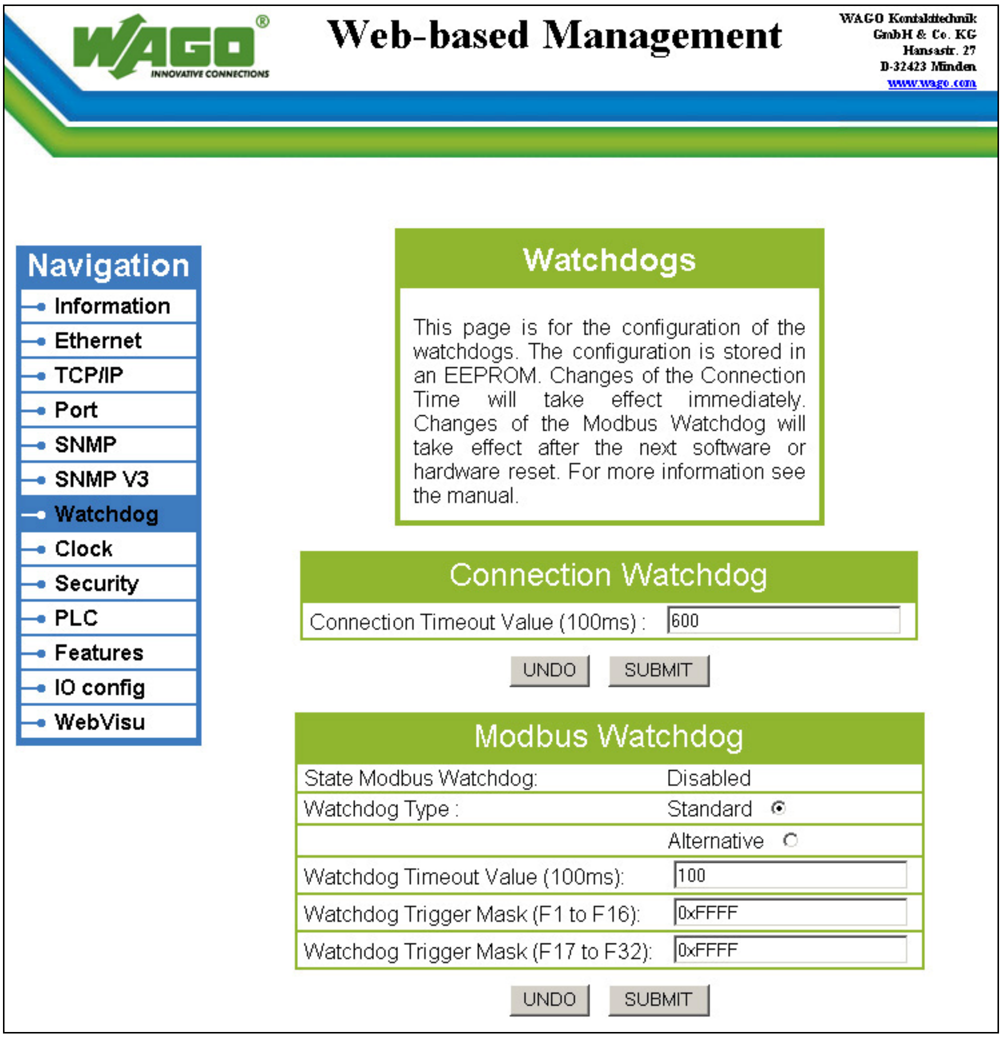


Figure 64: WBM page "Watchdog"

Table 44: WBM page "Watchdog"

Connection watchdog		
Entry	**Default**	**Description**
Connection Timeout Value (100 ms)	600	Monitoring period for TCP links. After the completion of this period without any subsequent data traffic, the TCP connection is closed.
Modbus Watchdog		
Entry	**Default**	**Description**
State Modbus Watchdog	Disabled	Enabled – Watchdog is activated Disabled – Watchdog is disabled
Watchdog Type	Standard ◉	The set coding mask (watchdog trigger mask) is evaluated to determine whether the watchdog time is reset.
	Alternative ○	The watchdog time is reset by any Modbus/TCP telegram.
Watchdog Timeout Value (100 ms)	100	Monitoring period for Modbus links. After the completion of this period without receiving a Modbus telegram, the physical outputs are set to "0".
Watchdog Trigger Mask (F 1 to F16)	0xFFFF	Coding mask for certain Modbus telegrams (Function Code FC1 ... FC16)
Watchdog Trigger Mask (F17 to F32)	0xFFFF	Coding mask for certain Modbus telegrams (Function Code FC17 ... FC32)

10.7 Clock

Specify the settings for the internal real-time clock on the "Clock" HTML page. Here, enter the current time and date and also select standard or daylight saving time.

Note

Reset the internal clock after 6 days without power supply!
The internal clock must be reset on initial startup or after 6 days without power. If the clock is not set, the clock begins with the date 01.01.2000 around 0:00 clock with time measurement.

Note

Integrate the function block for converting from winter/summer time!
Switch-over between standard and daylight saving time via Web-based management system is required when synchronizing the controllers in your network using a time server. The controller itself does not automatically execute a change-over between standard and daylight-saving time. The change-over is resolved via function block PrgDaylightSaving, which you must integrate into the WAGO-I/O-*PRO* using the library DaylightSaving.lib. From that point, change-over will be performed automatically, allowing all functions to be executed properly and at the right time.

Note

Error message in WAGO I/O *CHECK* is possible after a power failure!
If you are using the software "WAGO-I/O-*CHECK*" after a loss of power has occurred, error messages may be generated. Should this occur, call up the Web-based management system and set the actual time under "Clock". Then, call up the "WAGO-I/O-*CHECK*" program again

Note

Loss of telegrams possible when performing configuration during ongoing operation!
Telegrams may be lost if configuration is performed using WAGO-I/O-*CHECK* while the system is in operation.

Note

Use a WAGO RTC module for time synchronization!!
You can use a WAGO 750-640 RTC Module for your node to utilize the actual encoded time (Real-time – RTC) in your higher-level control system.

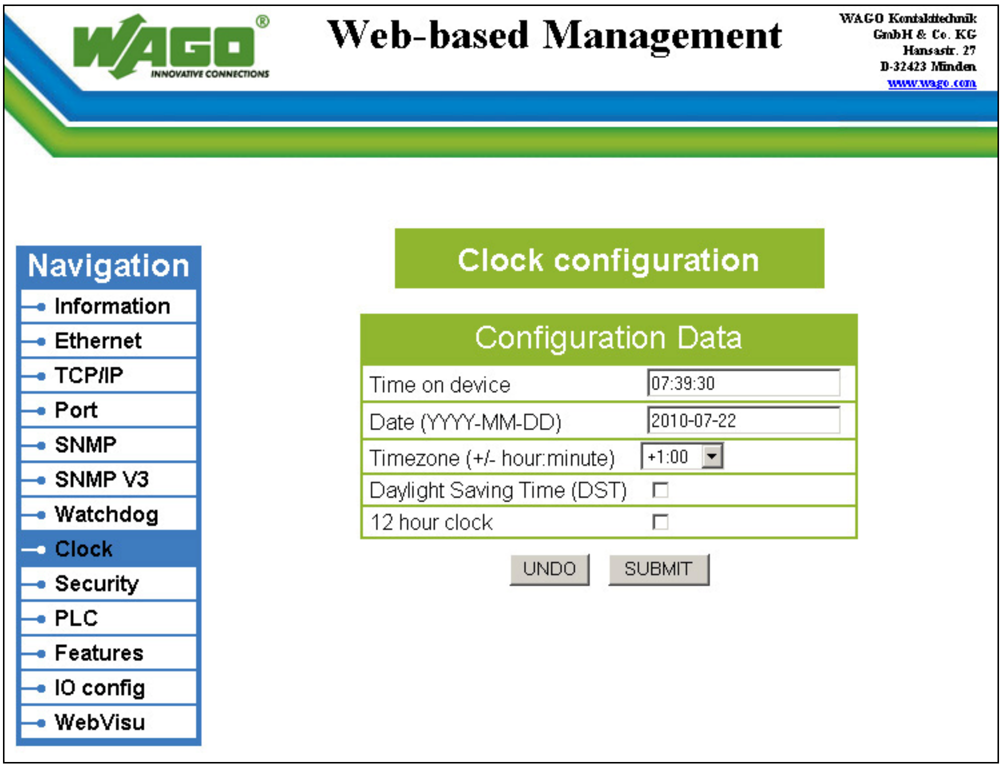


Figure 65: WBM page "Clock"

Table 45: WBM page "Clock"

Configuration Data			
Entry	**Default**	**Value (example)**	**Description**
Time on device	Coordinated Universal Time UTC	09:16:41	Set current time
Date (YYYY-MM-DD)	Date based on UTC	2009-05-06	Set current date
Time zone (+/- hour)	0	1 (MEZ)	Set time zone offset from the Coordinated Universal Time (UTC)
Daylight Saving Time (DST)/ Summer Time	Summer time ☑	Summer time ☑	☑ Enable summer time ☐ Enable winter time
12 hour clock	12 hour clock ☑	12 hour clock ☑	☑ Enable 12-hour display ☐ Enable 24-hour display

10.8 Security

Use the "Security" HTML page with passwords to set up read and/or write access for various user groups to protect against configuration changes.

Note

Passwords can only be changed by "admin" and after software reset!
The "admin" user and associated password are required to change passwords. Press the **[Software Reset]** button to restart the software for the setting changes to take effect.

Note

Note password restrictions!
The following restrictions apply for passwords:

- Max. 16 characters
- Letters and numbers only
- No special characters or umlauts

Note

Renew access after software reset!
If you initiate a software reset on this page, then the fieldbus coupler/controller starts with the configurations previously loaded into the EEPROM and the connection to the browser is interrupted.
If you changed the IP address previously, you have to use the changed IP address to access the device from the browser.
You have have not changed the IP address and performed other settings, you can restor the connection by refreshing the browser.

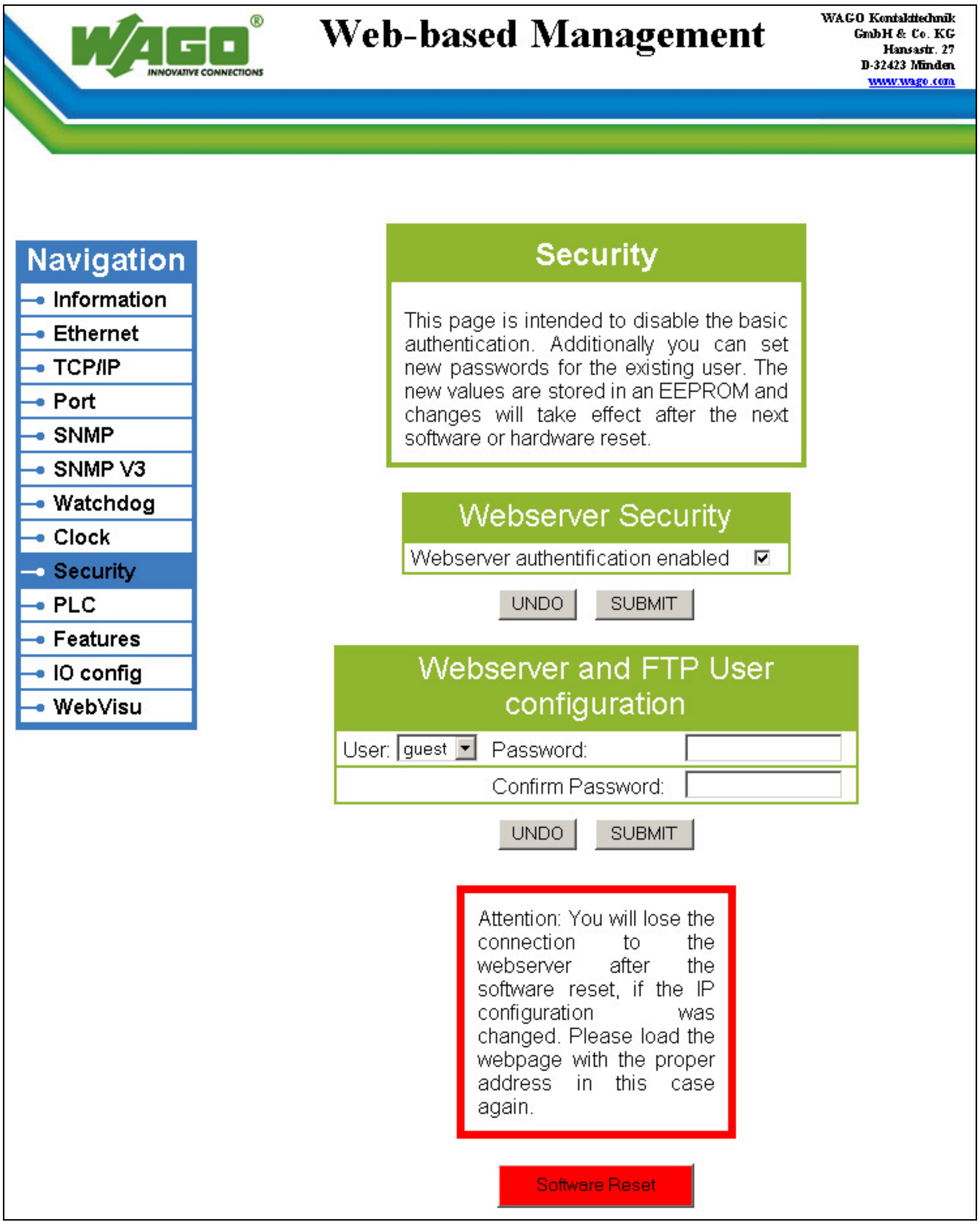


Figure 66: WBM page "Security"

Table 46: WBM page "Security"

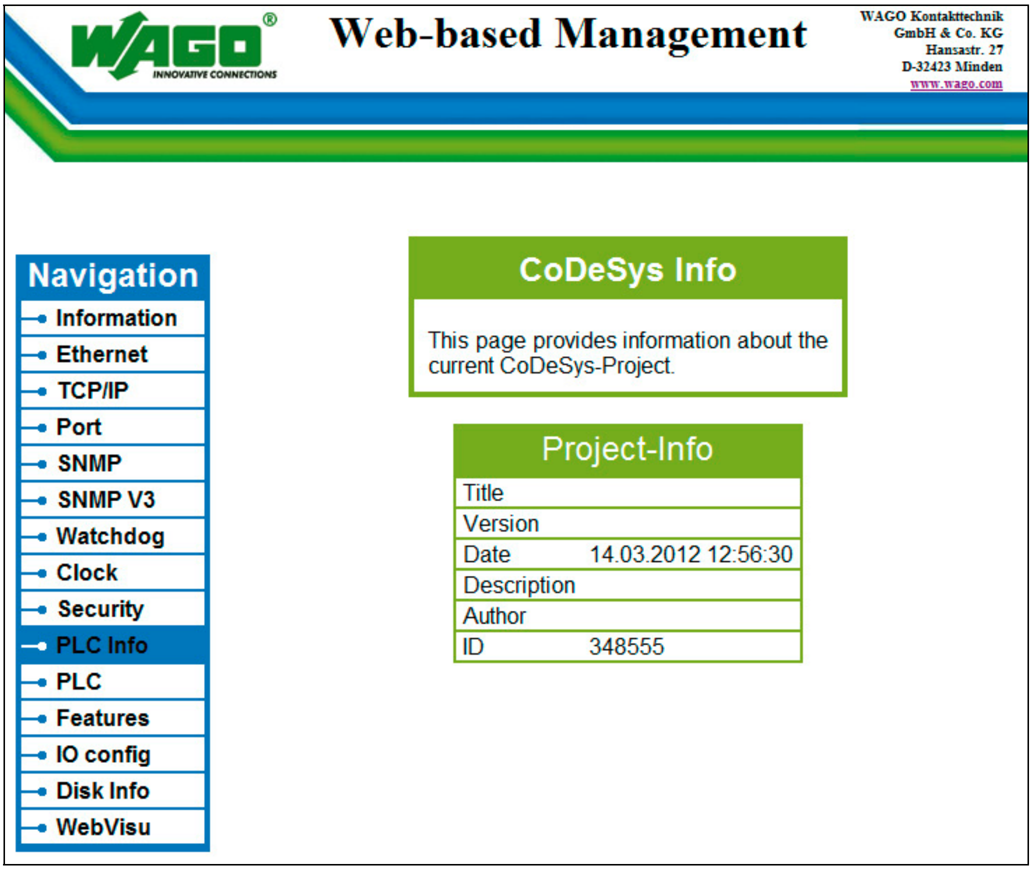
Webserver Security			
Entry	**Default**	**Description**	
Webserver authentification enabled	☑	☑	Enable password protection to access the Web interface
		☐	Disable password protection to access the Web interface
Webserver and FTP User configuration *)			
Entry	**Default**	**Description**	
User	guest	Select admin, guest or user	
Password	guest	Enter password	
Confirm password		Enter password again to confirm	

*) The following default groups exist:

User: admin	Password: wago
User: guest	Password: guest
User: user	Password: user

10.9 PLC Info

The WBM page „PLC Info “ contains information about the current CoDeSys project. First, these information has to be registered in the CoDeSys project under the menu „Project “→ „Projekt Information “.



Fugure 67: WBM page „PLC Info“

Table 47: WBM page „PLC Info“

Project Info			
Entry	**Default value**	**Value (Example)**	**Description**
Title	____	SSL Client Example	Name of project
Version	____	1.0.0	Project version
Date	____	15.05.2012 08:50:27	Date and Time of the project
Description	____	Testclient zum Aufbau von SSL-Verbindungen	Description
Author	____	JW	Project author
ID	____	70632	Project ID

10.10 PLC

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

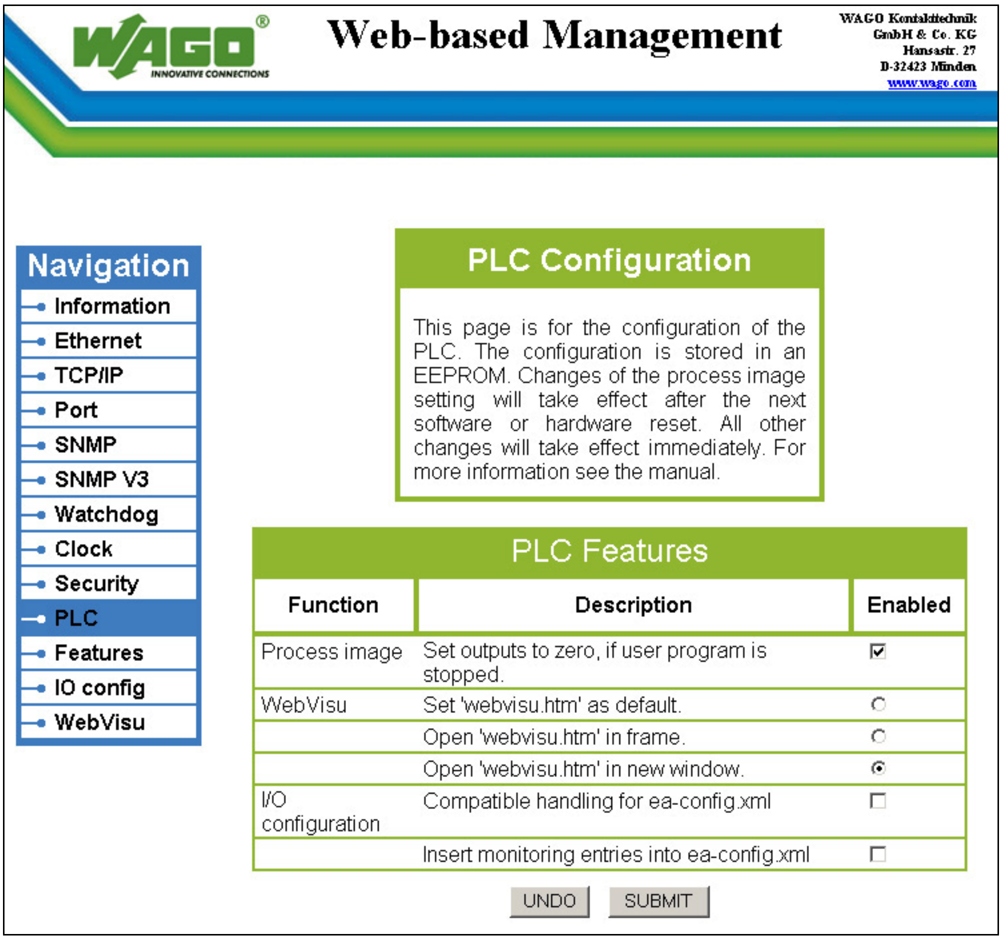


Figure 68: WBM page "PLC"

Note

Return to WBM view via the IP address of the fieldbus controller!
The "Webvisu.htm" page does not have any hyperlinks to the other Web sites. To deactivate this starting page function, or to go to other pages using hyperlinks, enter the IP address for your controller and the address for the original starting page in the URL line of your browser with the following syntax:
http://IP address of your controller/webserv/Index.ssi

Table 48: WBM page "PLC"

PLC Features			
Function	**Default**	**Description**	
Process image	Set outputs to zero, if user program is stopped ☐	☑	Activate, if all outputs must be set at zero when stopping the user program
		☐	Disable, if all outputs must remain at the last current value when stopping the user program
WebVisu	Set 'webvisu.htm' as default ○	◉	Activate, if the page "Webvisu.htm" must be opened as starting page when calling up WMB instead of the standard starting page "Status Information"
		○	Activate, if the standard starting page "Status Information" must be opened when calling up WMB
	Open 'webvisu.htm' in frame ○	◉	Activate, if the page "Webvisu.htm" must be opened in the same frame
		○	Activate, if the page "Webvisu.htm" must be opened in another frame
	Open 'webvisu.htm' in new window ◉	◉	Activate, if the page "Webvisu.htm" must be opened in the same window
		○	Activate, if the page "Webvisu.htm" must be opened in another window

Table: WBM page "PLC"

<table>
<tr><th colspan="5">PLC Features</th></tr>
<tr><th>Function</th><th>Default</th><th colspan="3">Description</th></tr>
<tr><td rowspan="8">I/O configu-ration</td><td rowspan="6">Compatible handling for ea-config.xml ☐</td><td colspan="3">☑ Activate, if the write authorizations must be assigned to the outputs of all bus terminals based on an existing file "ea-config.xml“.
Here, note whether a control system configuration has already been created and, if so, whether this configuration is correct or incorrect (see the following table).
The current process values are displayed on the website "IO config", in addition to the displayed data channels.</td></tr>
<tr><td colspan="3">☐ Disable, if the write authorizations must be assigned to the outputs of all bus terminals of the PLC
Here, note whether a control system configuration has already been created and, if so, whether this configuration is correct or incorrect (see the following table).</td></tr>
<tr><td></td><td>I/O configuration (function activated)</td><td>I/O configuration (function deactivated, standard setting):</td></tr>
<tr><td><u>No</u> control system configuration has been created in the project</td><td>Writing privileges to the outputs of all modules are assigned on the basis of an existing ea-config.xml.

The ea-config.xml file must be completely error-free; otherwise the writing privileges for all modules will be assigned to the standard fieldbus.</td><td>The outputs for all modules are assigned to the PLC. Any ea-config.xml file that may already be present is ignored and overwritten.</td></tr>
<tr><td><u>Correct</u> control system configuration has been created in the project</td><td colspan="2">Writing privileges to the module outputs is taken from the control system configuration. A corresponding ea-config.xml file is generated in the file system.</td></tr>
<tr><td><u>Incorrect</u> control system configuration has been created in the project</td><td colspan="2">The standard fieldbus is granted writing privileges to the outputs of all the modules.</td></tr>
<tr><td rowspan="2">Insert monitoring entries into ea-config.xml ☐</td><td colspan="3">☑ Activate to also display the current process values on the html page"IO config"
for the displayed data channels.</td></tr>
<tr><td colspan="3">☐ Disable, if no process values must be displayed on the html page "IO config".</td></tr>
</table>

10.11 Features

Use the "Features" HTML page to enable or disable additional functions.

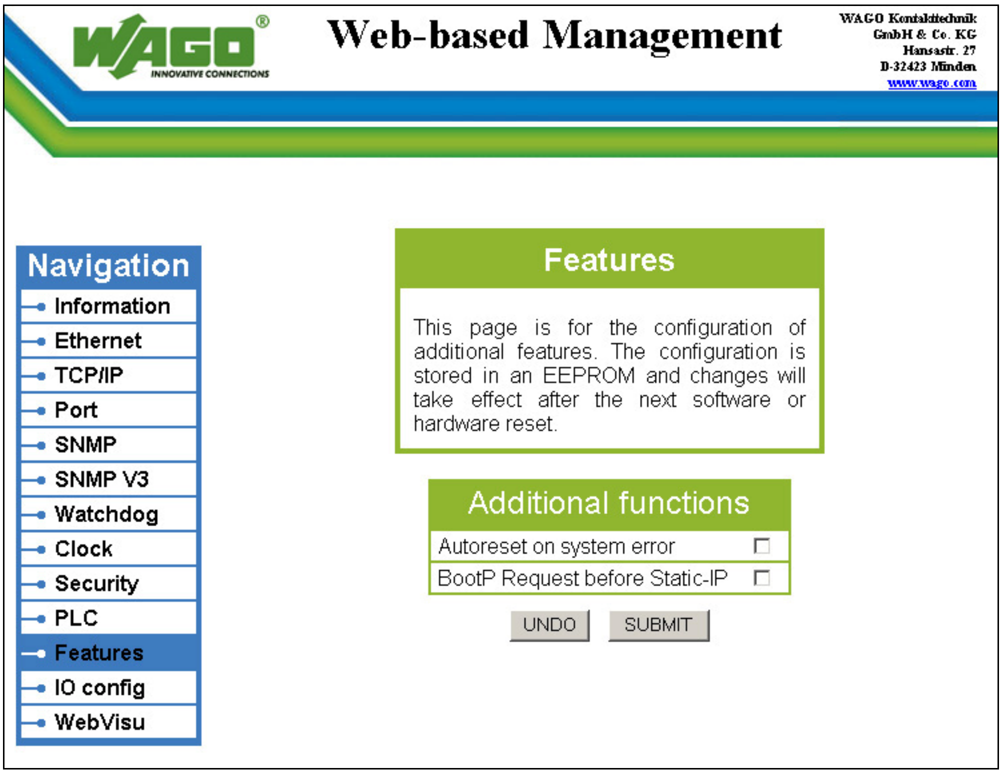


Figure 69: WBM page "Features"

Table 49: WBM page "Features"

Additional functions			
Entry	**Default**	**Description**	
Autoreset on system error	☐	☑	enables an automatic software reset to be conducted when a system error occurs
		☐	disables an automatic software reset to be conducted when a system error occurs
BootP Request before Static-IP	☐	☑	Automatically set the static IP address enabled. For this configuration, the fieldbus coupler/ controller uses a statically configured IP address if the request via BootP fails.
		☐	Automatically set the static IP address disabled. For this configuration, the IP address request via BootP is repeated in the event of error.

10.12 I/O Config

Click the link "I/O config" to view the configuration and/or write access privileges for the outputs of your fieldbus node.

The node structure created using the "WAGO-I/O-*PRO* I/O Configurator" hardware configuration tool is displayed in the window. If no modules are shown in this window, no hardware configuration and, thus, no allocation of write access privileges have been assigned. In this case, the handling defined at the Web site "PLC" by the function "I/O configuration - Compatible handling for ea-config.xml" will be applied to assign the write privileges for all outputs either to the standard fieldbus, or to the PLC.

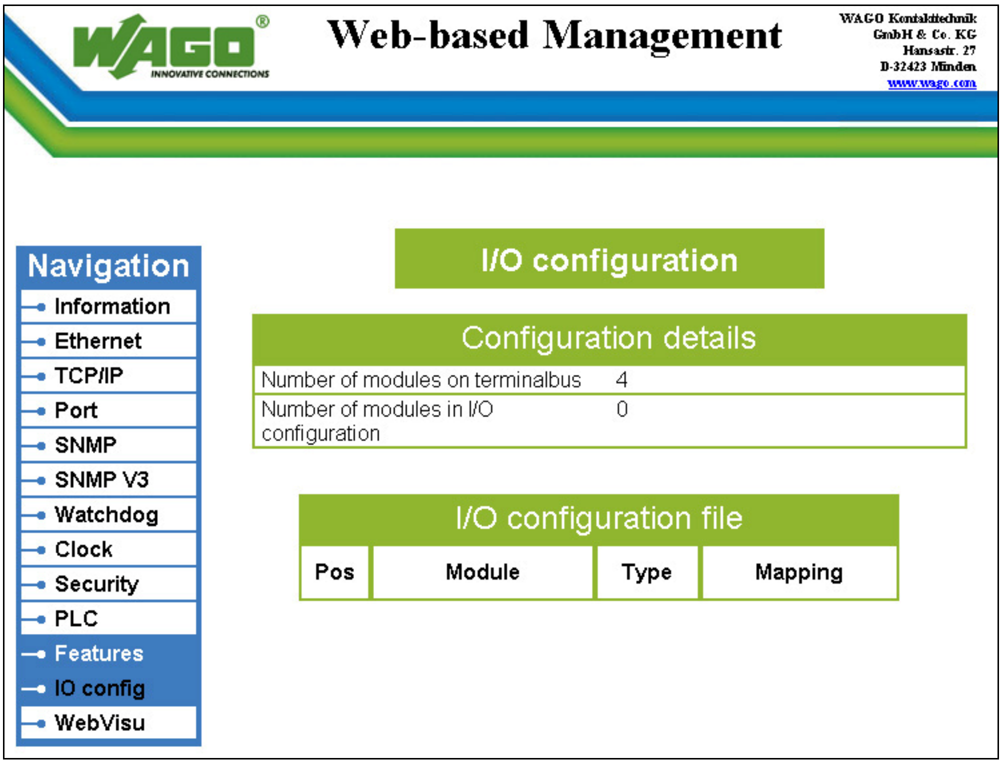


Figure 70: WBM page "I/O Config"

Information

Additional Information

For more detailed information about the WAGO-I/O-*PRO* I/O Configurator, refer to the Section "Startup of Fieldbus Node".

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

Table 50: WBM page "I/O configuration"

Configuration details		
Entry	**Value (Example)**	**Description**
Number of modules on terminalbus	5	Number of I/O modules (hardware)
Number of modules in I/O configuration	5	Number of I/O modules in the hardware configuration of the I/O

I/O configuration file		
Entry	**Value (Example)**	**Description**
Pos	1	Position of the I/O module in the hardware
Module	750-4xx M001Ch1 M001Ch2	Product number of the integrated I/O module M = module, 001 = position 1, Ch1 = channel 1 M = module, 002 = position 2, Ch2 = channel 2
Type	2DI	I/O module type, e.g. 2 DI (2 Channel Digital Input Module)
Mapping	Fieldbus 3	Mapping via PLC, fieldbus 1 etc. (Entries depend on the coupler/controller, see WAGO-I/O-*PRO* under control parameters/module parameters)

Note

Enter I/O modules in the I/O Configurator!
Enter the I/O modules used in the I/O configurator of WAGO-I/O-*PRO*. Here, open the **Control Configuration** in the **Resources** register and add your I/O modules to the I/O module figure.
The added I/O modules must match the hardware in sequence and quantity. The entries "Number of modules on terminalbus" and "Number of modules in I/O configuration" on the html page "PLC" serve as control.

10.13 WebVisu

The visualization of your programmed application is displayed on the html page "WebVisu", provided you have created it with the visualization editor in WAGO-I/O-*PRO* and loaded it into the controller.

Perform the following settings in WAGO-I/O-*PRO*, so that an html page with your visualization is automatically created at the transmission of your project:

1. Double click to open the **Target System Settings** in the **Resource** register.
2. Open the **Visualization** register.
3. Select the **Web Visualization** option with a hatch mark.
4. Confirm with **OK**.

A link is then created to this html page "WebVisu" by the Web-based Management system. You can set the html page "WebVisu" as the starting page.

1. Call up the page "PLC“ in the web-based Management-System.
2. a.) To set the HTML page "WebVisu" as the start page, use the function **WebVisu – Set 'webvisu.htm' as default**. When accessing the web-based management system, the "WebVisu“ page is opened instead of the default WBM start page "Information". However, the links to switch to the other WBM pages is then no longer available.

Note

Returning to the "WebVisu.htm" page is only possible via the IP address of the fieldbus controller!
The "Webvisu.htm" page does not have any hyperlinks to other Web sites. To deactivate the starting page function again, or to go to other pages using hyperlinks, enter the IP address for your controller and the address for the original starting page in the URL line of your browser with the following syntax:
http://IP address of your controller/webserv/Index.ssi.

b.) To call up the the HTML page "WebVisu" in an eternal window (default setting), use the function **WebVisu – Open 'webvisu.htm' in new window**. Clicking on the "WebVisu" link opens a new window that displays the HTML page with visualization of your configured application. The links to switch to the other WBM pages are still available with this setting.

c) To call up the HTML page "WebVisu" on the WBM site directly, use the function **WebVisu – Open 'webvisu.htm' in frame**. Clicking on the "WebVisu" link opens the HTML page with visualization of your configured application in a frame in the WBM window directly. The links to switch to the other WBM pages are still available with this setting.

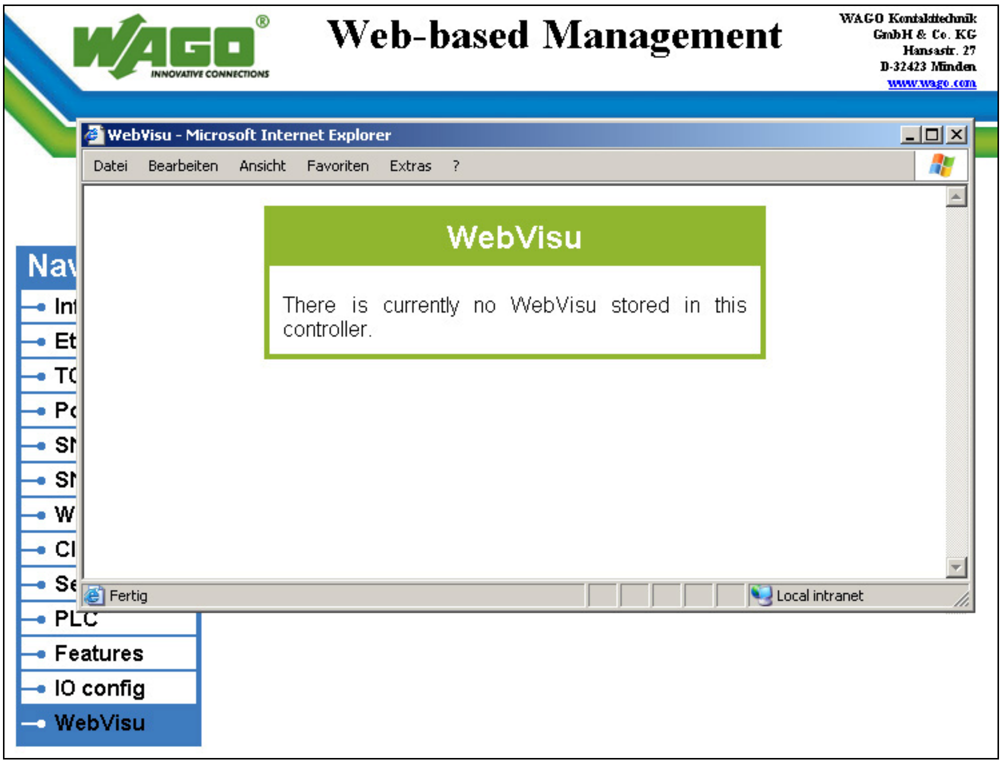


Figure 71: WBM page "WebVisu"

11 Diagnostics

11.1 LED Signaling

For on-site diagnostics, the fieldbus controller has several LEDs that indicate the operational status of the controller or the entire node (see following figure).

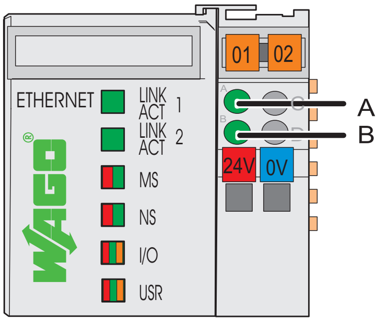


Figure 72: Display Elements

The diagnostics displays and their significance are explained in detail in the following chapter.

The LEDs are assigned in groups to the various diagnostics areas:

Table 51: LED assignment for diagnostics

Diagnostics area	LEDs
Fieldbus status	• LINK ACT Port 1 • LINK ACT Port 2 • MS • NS
Node status	• I/O • USR
Status Supply Voltage	• A (system supply) • B (field supply)

11.1.1 Evaluating Fieldbus Status

The health of the ETHERNET Fieldbus is signaled through the top LED group ('LINK ACT 1, 2', 'MS', und 'NS').

The two-colored LEDs ‘MS’ (module status) and ‘NS’ (network status) are solely used by the Ethernet/IP protocol. These two LEDs conform to the Ethernet/IP specifications.

Table 52: Fieldbus diagnostics – solution in event of error

LED Status	Meaning	Solution
LINK ACT 1, 2		
green	The fieldbus node is connected to the physical network.	-
green flashing	The fieldbus node sends and receives Ethernet telegrams	-
off	The fieldbus node is not connected to the physical network.	1. Check the fieldbus cable.
MS		
green	Normal operation	-
green flashing	The system is not yet configures	-
red	The system indicates a not remediable error	1. Restart the device by turning the power supply off and on again. 2. If the error still exists, please contact the I/O support.
red/green flashing	Self test	-
off	No system supply voltage	1. Check the supply voltage.
NS		
green	At least one connection (MODBUS/TCP or Ethernet/IP) is developed (also connection to the Message rout applies)	-
grün flashing	No connection (MODBUS/TCP or Ethernet/IP).	-
red	The system indicates a double IP-address in the network	1. Use an IP address that is not used yet.
red flashing	At least one connection (MODBUS/TCP or Ethernet/IP) announced a Timeout, where the controller functions as target.	1. Restart the device by turning the power supply off and on again. 2. Develop a new connection.
red/green flashing	Self test	-
off	No IP address is assigned to the system.	1. Assign to the system an IP address for example by BootP or DHCP.

11.1.2 Evaluating Node Status - I/O LED (Blink Code Table)

The communication status between fieldbus coupler/controller is indicated by the I/O LED.

Table 53: Node status diagnostics – solution in event of error

LED Status	Meaning	Solution
I/O		
green	The fieldbus node is operating correctly.	Normal operation.
orange flashing	The internal data bus is initialized, 1-2 seconds of rapid flashing indicate start-up.	-
red	Controller hardware defect	Replace the fieldbus coupler/controller.
red flashing	General internal bus error	Note the following blinking sequence.
red cyclical flashing	Up to three successive blinking sequences indicate internal data bus errors. There are short intervals between the sequences.	Evaluate the blinking sequences based on the following blink code table. The blinking indicates an error message comprised of an error code and error argument.
off	No data cycle on the internal bus.	The fieldbus coupler/controller supply is off.

Device boot-up occurs after turning on the power supply. The I/O LED is orange.

After a trouble-free start-up, the I/O LED is green.

In the event of an error, the I/O LED continues to blink red. Blink codes indicate detailed error messages. An error is indicated cyclically by up to 3 blinking sequences.

After elimination of the error, restart the node by turning the power supply of the device off and on again.

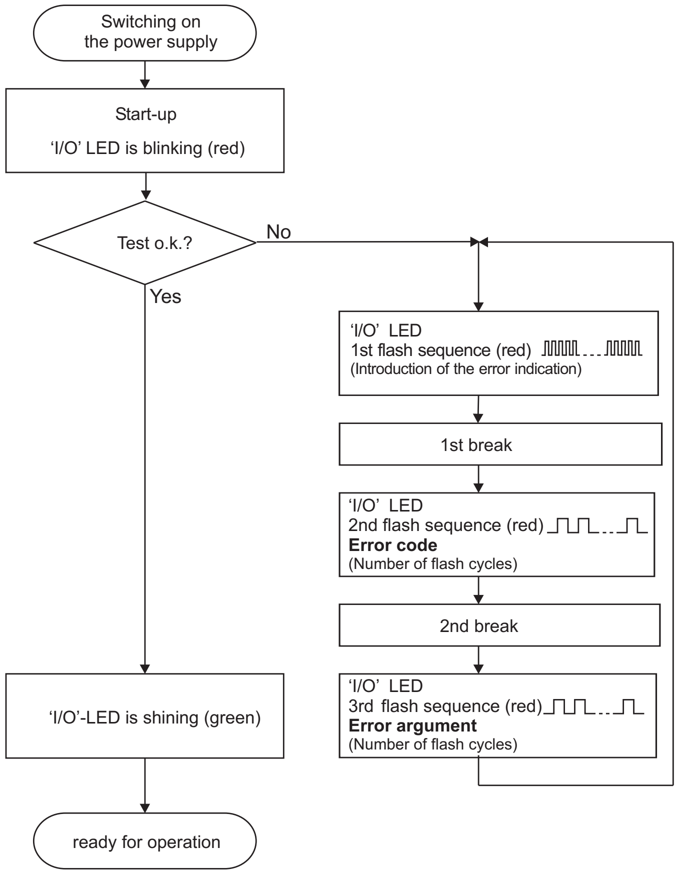


Figure 73: Node status - I/O LED signaling

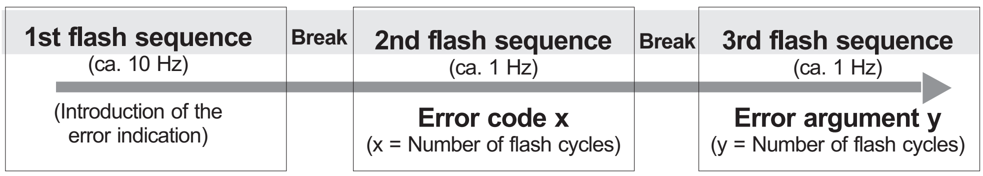


Figure 74: Error message coding

Example of a module error:

- The I/O LED starts the error display with the first blinking sequence (approx. 10 Hz).
- After the first break, the second blinking sequence starts (approx. 1 Hz):
  The I/O LED blinks four times.
  Error code 4 indicates "data error internal data bus".
- After the second break, the third blinking sequence starts (approx. 1 Hz):
  The I/O LED blinks twelve times.
  Error argument 12 means that the internal data bus is interrupted behind the twelfth I/O module.

The thirteenth I/O module is either defective or has been pulled out of the assembly.

Table 54: Blink code- table for the I/O LED signaling, error code 1

Error code 1: "Hardware and configuration error"		
Error Argument	**Error Description**	**Solution**
1	Overflow of the internal buffer memory for the attached I/O modules.	1. Turn off the power for the node. 2. Reduce the number of I/O modules and turn the power supply on again. 3. If the error persists, replace the fieldbus controller.
2	I/O module(s) with unknown data type	1. Determine the faulty I/O module by first turning off the power supply. 2. Plug the end module into the middle of the node. 3. Turn the power supply on again. 4. - LED continues to flash? - Turn off the power supply and plug the end module into the middle of the first half of the node (toward the fieldbus controller). - LED not flashing? - Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller). 5. Turn the power supply on again. 6. Repeat the procedure described in step 4 while halving the step size until the faulty I/O module is detected. 7. Replace the faulty I/O module. 8. Inquire about a firmware update for the fieldbus controller.
3	Invalid check sum in the parameter area of the fieldbus controller.	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
4	Fault when writing in the serial EEPROM.	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
5	Fault when reading the serial EEPROM	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
6	The I/O module configuration after AUTORESET differs from the configuration determined the last time the fieldbus controller was powered up.	1. Restart the fieldbus controller by turning the power supply off and on.
7	Invalid hardware-firmware combination.	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.

Table 54: Blink code- table for the I/O LED signaling, error code 1

Error code 1: "Hardware and configuration error"		
Error Argument	**Error Description**	**Solution**
8	Timeout during serial EEPROM access.	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
9	Bus controller initialization error	1. Turn off the power supply for the node. 2. Replace the fieldbus controller. 3. Turn the power supply on again.
10	Buffer power failure real-time clock (RTC)	1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
11	Fault during read access to the real-time clock (RTC)	1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
12	Fault during write access to the real-time clock (RTC)	1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
13	Clock interrupt fault	1. Set the clock. 2. Maintain the power supply of the fieldbus controller for at least 15 minutes in order to charge the Goldcap capacitor.
14	Maximum number of gateway or mailbox modules exceeded	1. Turn off the power for the node. 2. Reduce the number of corresponding modules to a valid number.

Table 55: Blink code table for the I/O LED signaling, error code 2

Error code 2: -not used-		
Error Argument	**Error Description**	**Solution**
-	Not used	-

Table 56: Blink code table for the I/O LED signaling, error code 3

Error code 3: "Protocol error, internal bus"		
Error Argument	**Error Description**	**Solution**
-	Internal data bus communication is faulty, defective module cannot be identified.	- Are passive power supply modules (750-613) located in the node? - 1. Check that these modules are supplied correctly with power. 2. Determine this by the state of the associated status LEDs. - Are all modules connected correctly or are there any 750-613 Modules in the node? - 1. Determine the faulty I/O module by turning off the power supply. 2. Plug the end module into the middle of the node. 3. Turn the power supply on again. 4. - LED continues to flash? - Turn off the power supply and plug the end module into the middle of the first half of the node (toward the fieldbus controller). - LED not flashing? - Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller). 5. Turn the power supply on again. 6. Repeat the procedure described in step 4 while halving the step size until the faulty I/O module is detected. 7. Replace the faulty I/O module. 8. Inquire about a firmware update for the fieldbus controller.

Table 57: Blink code table for the I/O LED signaling, error code 4

Error code 4: "Physical error, internal bus"		
Error Argument	**Error Description**	**Solution**
-	Internal bus data transmission error or interruption of the internal data bus at the fieldbus controller	1. Turn off the power supply to the node. 2. Plug in an end module behind the fieldbus controller. 3. Turn the power supply on. 4. Observe the error argument signaled. - Is no error argument indicated by the I/O LED? - 5. Replace the fieldbus controller. - Is an error argument indicated by the I/O LED? - 5. Identify the faulty I/O module by turning off the power supply. 6. Plug the end module into the middle of the node. 7. Turn the power supply on again. 8. - LED continues to flash? - Turn off the power and plug the end module into the middle of the first half of the node (toward the fieldbus controller). - LED not flashing? - Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller). 9. Turn the power supply on again. 10. Repeat the procedure described in step 6 while halving the step size until the faulty I/O module is detected. 11. Replace the faulty I/O module. 12. If there is only one I/O module on the fieldbus controller and the LED is flashing, either the I/O module or fieldbus controller is defective. Replace the defective component.
n*	Interruption of the internal data bus behind the nth bus module with process data	1. Turn off the power supply to the node. 2. Replace the (n+1) I/O module containing process data. 3. Turn the power supply on.

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

Table 58: Blink code table for the I/O LED signaling, error code 5

Error code 5: "Initialization error, internal bus"		
Error Argument	**Error Description**	**Solution**
n*	Error in register communication during internal bus initialization	1. Turn off the power supply to the node. 2. Replace the (n+1) I/O module containing process data. 3. Turn the power supply on.

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

Table 59: Blink code- table for the I/O LED signaling, error code 6

Error code 6: "Fieldbus specific errors"		
Error Argument	**Error description**	**Solution**
1	Invalid MACID	1. Turn off the power supply of the node. 2. Exchange fieldbus controller. 3. Turn the power supply on again.
2	Ethernet Hardware initialization error	1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, exchange the fieldbus controller.
3	TCP/IP initialization error	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. If the error still exists, exchange the bus coupler.
4	Network configuration error (no IP Address)	1. Check the settings of BootP server.
5	Application protocol initialization error	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. If the error still exists, exchange the bus coupler.
6	Process image is too large	1. Turn off the power supply of the node. 2. Reduce number of I/O modules
7	Double IP address in network	1. Change configuration. Use another IP address, which is not yet present in network. 2. Restart the fieldbus coupler by turning the power supply off and on again.
8	Error when building the process image	1. Turn off the power supply of the node. 2. Reduce number of I/O modules 3. Restart the fieldbus coupler by turning the power supply off and on again. 4. If the error still exists, exchange the bus coupler.
9	Error with mapping between bus modules and fieldbus	1. Check EA-Config.xml file on the fieldbus controller

Table 60: Blink code table for the 'I/O' LED signaling, error code 7…9

Error code 7…9: -not used-		
Error Argument	**Error Description**	**Solution**
-	Not used	

Table 61: Blink code table for the 'I/O' LED signaling, error code 10

Error code 10: "PLC program fault"		
Error Argument	**Error Description**	**Solution**
1	Error when implementing the PFC run time system	1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, please contact the I/O Support.
2	Error when generating the PFC inline code	1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, please contact the I/O Support.
3	An IEC task exceeded the maximum running time or the sampling interval of the IEC task could not be kept (Watchdog)	1. Check the task configuration concerning the adjusted sampling intervals and watchdog times.
4	PFC Web-Visualization initialization error	1. Restart the fieldbus controller by turning the power supply off and on again. 2. If the error still exists, please accomplish a reset (origin) in WAGO-I/O-*PRO*. 3. Compile the project again. 4. Transfer the project to the controller.
5	Error when synchronizing the PLC configuration with the internal data bus	1. Check the information of the connected modules in the PLC configuration of WAGO-I/O-*PRO*. 2. Compare this information with the modules that are actually connected. 3. Compile the project again. 4. Transfer the project to the controller.

Table 62: Blink code table for the 'I/O' LED signaling, error code 11

Error code 11: "Gateway-/Mailbox I/O module fault"		
Error Argument	**Error Description**	**Solution**
1	Maximum number of Gateway modules exceeded	1. Turn off the power supply of the node. 2. Reduce number of Gateway modules. 3. Turn the power supply on again.
2	Maximum size of Mailbox exceeded	1. Turn off the power supply of the node. 2. Reduce the Mailbox size. 3. Turn the power supply on again.
3	Maximum size of process image exceeded due to the put Gateway modules	1. Turn off the power supply of the node. 2. Reduce the data width of the Gateway modules. 3. Turn the power supply on again.

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

11.1.2.1 USR LED

The bottom indicator LED ("USR") is provided for visual output of information.

Control of the LED from the application program is conducted using the functions from the WAGO-I/O-PRO library "Visual.lib."

11.1.3 Evaluating Power Supply Status

The power supply unit of the device has two green LEDs that indicate the status of the power supply.
LED 'A' indicates the 24 V supply of the coupler.
LED 'B' or 'C' reports the power available on the power jumper contacts for field side power.

Table 63: Power supply status diagnostics – solution in event of error

LED Status	Meaning	Solution
A		
Green	Operating voltage for the system is available.	-
Off	No power is available for the system	Check the power supply for the system (24V and 0V).
B or C		
Green	The operating voltage for power jumper contacts is available.	-
Off	No operating voltage is available for the power jumper contacts.	Check the power supply for the power jumper contacts (24V and 0V).

11.2 Fault Behavior

11.2.1 Loss of Fieldbus

A fieldbus and, hence, a link failure is recognized when the set reaction time for the watchdog expires without initiation by the higher-order control system. This may occur, for example, when the Master is switched off, or when there is a disruption in the bus cable. An error at the Master can also result in a fieldbus failure. No connection via ETHERNET.

The MODBUS watchdog monitors the ongoing MODBUS communication via MODBUS protocol. A fieldbus failure is signaled by the red "I/O" LED lighting up, provided the MODBUS watchdog has been configured and activated.

Fieldbus monitoring independently of a certain protocol is possible using the function block 'FBUS_ERROR_INFORMATION' in the library "Mod_com.lib". This checks the physical connection between modules and the controller and assumes evaluation of the watchdog register in the control system program. The I/O bus remains operational and the process images are retained. The control system program can also be processed independently.

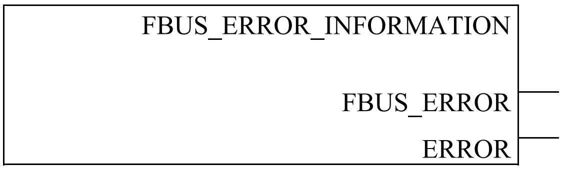


Figure 75: Function block for determining loss of fieldbus, independently of protocol

'FBUS_ERROR' (BOOL)	= FALSE	= no fault
	= TRUE	= loss of field bus
'ERROR' (WORD)	= 0	= no fault
	= 1	= loss of field bus

The node can be put into a safe status in the event of a fieldbus failure with the aid of these function block outputs and an appropriately programmed control system program.

Information

Loss of fieldbus detection through MODBUS protocol:
For detailed information about the watchdog register, refer to Section "MODBUS Functions", in particular Section "Watchdog (Fieldbus failure)".

Protocol-independent detection of loss of fieldbus:
The library 'Mod_com.lib' with function block 'FBUS_ERROR_INFORMATION' is normally included in the setup for the WAGO-I/O-PRO. You can integrate the library via register "Resources" at the bottom on the left of the workspace. Click **Insert** and then **Other libraries**. The Mod_com.lib is located in folder C:\Programme\ WAGO Software\CoDeSys V2.3\Targets\WAGO\Libraries\32_Bit

11.2.2 Internal Data Bus Failure

I/O LED indicates an internal bus failure.

I/O LED flashed red:
When an internal data bus failure occurs, the fieldbus controller generates an error message (error code and error argument).
An internal data bus failure occurs, for example, if an I/O module is removed.
If the error occurs during operation, the output modules operate as they do during an internal data bus stop.
If the internal data bus error is resolved, the controller starts up after turning the power off and on similar to that of a normal start-up. The process data is transmitted again and the outputs of the node are set accordingly.

If the 'KBUS_ERROR_INFORMATION' function block is evaluated in the control program, then the 'ERROR','BITLEN', 'TERMINALS' and 'FAILADDRESS' output values are relevant.

'ERROR'	= FALSE	= No fault
	('BITLEN'	= Bit length of the internal bus shift register
	'TERMINALS'	= Number of I/O modules)

'ERROR'	= TRUE	= Internal Bus Error
	('BITLEN'	= 0
	'TERMINALS'	= 0)
	'FAILADRESS'	= Position of the I/O module after which the internal bus interruption arose, similar to the flashed error argument of the I/O LED

12 Fieldbus Communication

Fieldbus communication between master application and a WAGO fieldbus coupler/controller based on the ETHERNET standard normally occurs via an implemented fieldbus-specific application protocol.

Depending on the application, this can be e.g., MODBUS/TCP (UDP), EtherNet/IP, BACnet/IP, KNXnet/IP, PROFINET, SERCOS III or other.

In addition to the ETHERNET standard and the fieldbus-specific application protocol, there are also other communications protocols important for reliable communication and data transmission and other related protocols for configuring and diagnosing the system implemented in the WAGO fieldbus coupler/controller based on ETHERNET.

These protocols are explained in more detail in the other sections.

12.1 Implemented Protocols

12.1.1 Communication Protocols

12.1.1.1 IP (Internet Protocol)

The Internet protocol divides datagrams into segments and is responsible for their transmission from one network subscriber to another. The stations involved may be connected to the same network or to different physical networks which are linked together by routers.
Routers are able to select various paths (network transmission paths) through connected networks, and bypass congestion and individual network failures. However, as individual paths may be selected which are shorter than other paths, datagrams may overtake each other, causing the sequence of the data packets to be incorrect.
Therefore, it is necessary to use a higher-level protocol, for example, TCP to guarantee correct transmission.

IP Packet

In addition to the data units to be transported, the IP data packets contain a range of address information and additional information in the packet header.

Table 64: IP Packet

IP Header	IP Data

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

IP Addresses

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

Note

IP Address must be unique!
For error free operation, the IP address must be unique within the network.

As shown below there are various address classes with net identification (net ID) and subscriber identification (subscriber ID) of varying lengths. The net ID defines the network in which the subscriber is located. The subscriber ID identifies a particular subscriber within this network.

Networks are divided into various network classes for addressing purposes:

- **Class A:** (Net ID: Byte 1, Host ID: Byte 2… Byte 4)

Table 65: Network Class A

e. g. 101 . 16 . 232 . 22

<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’. This means the highest byte can be in a range of’0 0000000’ to ‘0 1111111’.
Therefore, the address range of a Class A network in the first byte is always between 0 and 127.

- **Class B:** (Net ID: Byte 1 … Byte 2, Host ID: Byte 3… Byte 4)

Table 66: Network Class B

e. g. 181 . 16 . 232 . 22

<table>
<tr><td colspan="2">10110101</td><td>00010000</td><td>11101000</td><td>00010110</td></tr>
<tr><td>10</td><td colspan="2">Net ID</td><td colspan="2">Host ID</td></tr>
</table>

The highest bits in Class B networks are always ’10’. This means the highest byte can be in a range of’10 000000’ to ‘10 111111’.
Therefore, the address range of Class B networks in the first byte is always between 128 and 191.

- **Class C:** (Net ID: Byte 1 … Byte 3, Host ID: Byte 4)

Table 67: Network Class C

e. g. 201 . 16 . 232 . 22

<table>
<tr><td colspan="2">11000101</td><td>00010000</td><td>11101000</td><td>00010110</td></tr>
<tr><td>110</td><td colspan="3">Net ID</td><td>Host ID</td></tr>
</table>

The highest bits in Class C networks are always ‘110’. This means the highest byte can be in a range of’110 00000’ to ‘110 11111’.

Therefore, the address range of Class C networks in the first byte is always between 192 and 223.

- **Additional network classes (D, E):** are only used for special tasks.

Key Data

Table 68: Key Data Class A, B and C

Network Class	Address range of the subnetwork	Possible number of Networks	Possible number of Hosts per Network
Class A	1.XXX.XXX.XXX ... 126.XXX.XXX.XXX	127 (2^7)	Approx. 16 Million (2^{24})
Class B	128.000.XXX.XXX ... 191.255.XXX.XXX	Approx. 16 Thousand (2^{14})	Ca. 65 Thousand (2^{16})
Class C	192.000.000.XXX ... 223.255.255.XXX	Approx. 2 Million (2^{21})	254 (2^8)

Each WAGO ETHERNET fieldbus coupler or controller can be easily assigned an IP address via the implemented BootP protocol. For small internal networks we recommend selecting a network address from Class C.

Note

Do not set IP addresses to 0.0.0.0 or 255.255.255.255!
Never set all bits to equal 0 or 1 in one byte (byte = 0 or 255). These are reserved for special functions and may not be allocated. Therefore, the address 10.0.10.10 may not be used due to the 0 in the second byte.

If a network is to be directly connected to the Internet, only registered, internationally unique IP addresses allocated by a central registration service may be used. These are available from Inter*NIC* (International Network Information Center).

Note

Internet access only by the authorized network administrator!
Direct connection to the Internet should only be performed by an authorized network administrator and is therefore not described in this manual.

Subnets

To allow routing within large networks a convention was introduced in the specification RFC 950. Part of the Internet address, the subscriber ID is divided up again into a subnetwork number and the station number of the node. With the aid of the network number it is possible to branch into internal subnetworks within the partial network, but the entire network is physically connected together. The size and position of the subnetwork ID are not defined; however, the size is dependent upon the number of subnets to be addressed and the number of subscribers per subnet.

Table 69: Class B Address with Field for Subnet IDs

1			8	16	24	32
1	0	...	Network ID	Subnet ID	Host ID	

Subnet Mask

A subnet mask was introduced to encode the subnets in the Internet. This involves a bit mask, which is used to mask out or select specific bits of the IP address. The mask defines the subscriber ID bits used for subnet coding, which denote the ID of the subscriber. The entire IP address range theoretically lies between 0.0.0.0 and 255.255.255.255. Each 0 and 255 from the IP address range are reserved for the subnet mask.

The standard masks depending upon the respective network class are as follows:

- **Class A Subnet mask:**

Table 70: Subnet mask for Class A network

255	.0	.0	.0

- **Class B Subnet mask:**

Table 71: Subnet mask for Class B network

255	.255	.0	.0

- **Class C Subnet mask:**

Table 72: Subnet mask for Class C network

255	.255	.255	.0

Depending on the subnet division the subnet masks may, however, contain other values beyond 0 and 255, such as 255.255.255.128 or 255.255.255.248.

Your network administrator allocates the subnet mask number to you.

Together with the IP address, this number determines which network your PC and your node belongs to.

The recipient node, which is located on a subnet initially, calculates the correct network number from its own IP address and the subnet mask.

Only then does it check the node number and delivers the entire packet frame, if it corresponds.

Table 73: Example for an IP address from a Class B network

IP address	172.16.233.200	10101100 00010000 11101001 11001000
Subnet mask	255.255.255.128	11111111 11111111 11111111 10000000
Net ID	172.16.0.0	10101100 00010000 00000000 00000000
Subnet ID	0.0.233.128	00000000 00000000 11101001 10000000
Host ID	0.0.0.72	00000000 00000000 00000000 01001000

Note

Specification of the network mask necessarily!
Specify the network mask defined by the administrator in the same way as the IP address when installing the network protocol.

Gateway

The subnets of the Internet are normally connected via gateways. The function of these gateways is to forward packets to other networks or subnets.

This means that in addition to the IP address and network mask for each network card, it is necessary to specify the correct IP address of the standard gateway for a PC or fieldbus node connected to the Internet. You should also be able to obtain this IP address from your network administrator.

The IP function is limited to the local subnet if this address is not specified.

RAW IP

Raw IP manages without protocols such as PPP (point-to-point protocol). With RAW IP, the TCP/IP packets are directly exchanged without handshaking, thus enabling the connection to be established more quickly.

However, the connection must beforehand have been configured with a fixed IP address. The advantages of RAW IP are high data transfer rate and good stability.

IP Multicast

Multicast refers to a method of transmission from a point to a group, which is a point-to-multipoint transfer or multipoint connection. The advantage of multicast is that messages are simultaneously transferred to several users or closed user groups via one address.

IP multicasting at the Internet level is realized with the help of the Internet Group Message Protocol IGMP; neighboring routers use this protocol to inform each other on membership to the group.

For distribution of multicast packets in the sub-network, IP assumes that the datalink layer supports multicasting. In the case of Ethernet, you can provide a packet with a multicast address in order to send the packet to several recipients with a single send operation. Here, the common medium enables packets to be sent simultaneously to several recipients. The stations do not have to inform each other on who belongs to a specific multicast address – every station physically receives every packet. The resolution of IP address to Ethernet address is solved by the use of algorithms, IP multicast addresses are embedded in Ethernet multicast addresses.

12.1.1.2 TCP (Transmission Control Protocol)

As the layer above the Internet protocol, TCP (Transmission Control Protocol) guarantees the secure transport of data through the network.

TCP enables two subscribers to establish a connection for the duration of the data transmission. Communication takes place in full-duplex mode (i.e., transmission between two subscribers in both directions simultaneously).

TCP provides the transmitted message with a 16-bit checksum and each data packet with a sequence number.

The receiver checks that the packet has been correctly received on the basis of the checksum and then sets off the sequence number. The result is known as the acknowledgement number and is returned with the next self-sent packet as an acknowledgement.

This ensures that the lost TCP packets are detected and resent, if necessary, in the correct sequence.

TCP Data Packet

The packet header of a TCP data packet is comprised of at least 20 bytes and contains, among others, the application port number of the transmitter and the receiver, the sequence number and the acknowledgement number.

The resulting TCP packet is used in the data unit area of an IP packet to create a TCP/IP packet.

TCP Port Numbers

TCP can, in addition to the IP address (network and subscriber address), respond to a specific application (service) on the addressed subscriber. For this the applications located on a subscriber, such as a web server, FTP server and others are addressed via different port numbers. Well-known applications are assigned fixed ports to which each application can refer when a connection is built up (Examples: Telnet Port number: 23, http Port number: 80).
A complete list of "standardized services" is contained in the RFC 1700 (1994) specifications.

12.1.1.3 UDP (User Datagram Protocol)

The UDP protocol, like the TCP protocol, is responsible for the transport of data. Unlike the TCP protocol, UDP is not connection-orientated; meaning that there are no control mechanisms for the data exchange between transmitter and receiver. The advantage of this protocol is the efficiency of the transmitted data and the resulting higher processing speed.

12.1.2 Configuration and Diagnostics Protocols

12.1.2.1 BootP (Bootstrap Protocol)

The "Bootstrap Protocol" (BootP) can be used to assign an IP address and other parameters to the fieldbus coupler/controller in a TCP/IP network. Subnet masks and gateways can also be transferred using this protocol. Protocol communication is comprised of a client request from the fieldbus coupler or controller and a server response from the PC.

A broadcast request is transmitted to Port 67 (BootP server) via the protocol that contains the hardware address (MAC ID) for the fieldbus coupler or controller.

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

The fieldbus coupler/controller "listens" at the specified Port 68 for a response from the BootP server. Incoming packets contain information such as the IP address and the MAC address for the fieldbus coupler/controller. A fieldbus coupler/controller recognizes by the MAC address that the message is intended for that particular fieldbus coupler/controller and accepts the transmitted IP address into its network.

Note

IP addresses can be assigned via BootP under Windows and Linux!
You can use WAGO-BootP-Server to assign an IP address under the Windows and Linux operating systems. You can also use any other BootP server besides WAGO-BootP-Server. You can also use any other BootP server besides the WAGO-BootP-Server.

Information

More information about WAGO-BootP-Server
The process for assigning addresses using WAGO-BootP-Server is described in detail in the section "Commissioning Fieldbus Node".

The BootP Client assists in the dynamic configuration of the network parameters: The ETHERNET TCP/IP fieldbus controller has a BootP client that supports the following options in addition to the default "IP address" option:

Table 74: BootP options

Option	Meaning
[OPT1] Subnet mask	32-bit address mask that displays which bits of the IP address identify the network and which identify the network stations.
[OPT2] Time zone	Time difference between the local time and the UTC (Universal Time Coordinated).
[OPT3] Gateway	IP address of the router that permits access to other networks.
[OPT6] DNS server	IP address of the name servers that converts a name into an IP address. Up to 2 DNS servers can be configured.
[OPT12] Host name	The name of the host is the unique name of a computer in a network. The host name can contain up to 32 characters.
[OPT15] Domain name	The name of the domain is the unique name of a network. The domain name can contain up to 32 characters.
[OPT42] NTP server	IP address of the Network Time Server. When assigning an NTP server, the SNTP client is automatically enabled in the coupler.

The "Features" WBM page can also be used to select the "BootP Request before static IP" option. After the restart, 5 BootP queries are sent. If there is no response to any of these queries, the fieldbus coupler/controller tries to configure itself with the IP parameters saved in the EEPROM.

The network parameters (IP address, etc.) are stored in the EEPROM when using the Bootstrap protocol to configure the node.

Note

BootP configuration is saved in the EEPROM!
Please note that the network configuration is stored in the EEPROM when using BootP in contrast to configuration via DHCP.

By default, BootP is activated in the fieldbus coupler/controller.

When BootP is activated, the fieldbus coupler/controller expects the BootP server to be permanently available.
If there is no BootP server available after a PowerOn reset, the network will remain inactive.

To operate the fieldbus coupler/controller with the IP configuration stored in the EEPROM, you must deactivate the BootP protocol after configuration.
The Web-based management system is used to deactivate the BootP protocol on the respective fieldbus coupler/controller-internal HTML page under the "Port" link.

If BootP is deactivated, the fieldbus coupler/controller uses the parameters saved in the EEPROM when booting next.

If there is an error in the saved parameters, the I/O LED reports a blink code and configuration via BootP is turned on automatically.

12.1.2.2 DHCP (Dynamic Host Configuration Protocol)

The fieldbus coupler/controller internal HTML page opened via the "Port" link provides the option to configure the network using the data saved in the EEPROM or via DHCP instead of via the BootP protocol.

DHCP (Dynamic Host Configuration Protocol) is a further development of BootP and is backwards compatible with BootP.
Both BOOTP and DHCP assign an IP address to the fieldbus node (Client) when starting; the sequence is the same as for BootP.

For configuration of the network parameters via DHCP, the fieldbus coupler/controller sends a client request to the DHCP server e.g., on the connected PC.

A broadcast request is transmitted to Port 67 (DHCP server) via the protocol that contains the hardware address (MAC ID) for the fieldbus coupler/controller.

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

The fieldbus coupler/controller "listens" at the specified Port 68 for a response from the DHCP server. Incoming packets contain information such as the IP address and the MAC address for the fieldbus coupler/controller. A fieldbus coupler/controller recognizes by the MAC address that the message is intended for that particular fieldbus coupler/controller and accepts the transmitted IP address into its network.

If there is no reply, the inquiry is sent again after 4 seconds, 8 seconds and 16 seconds.
If all inquiries receive no reply, a blink code is reported via the I/O LED. The parameters cannot be applied from the EEPROM.

Note

DHCP configuration is not saved in the EEPROM!
Please note that the network configuration is not stored in the EEPROM when using DHCP in contrast to configuration via BootP.

The difference between BOOTP and DHCP is that both use different assignment methods and that configuration with DHCP is time limited. The DHCP client always has to update the configuration after the time has elapsed. Normally, the same parameters are continuously confirmed by the server.

The difference between BOOTP and DHCP is that both use different assignment methods. BOOTP can be used to assign a fixed IP address for each client where the addresses and their reservation are permanently saved in the BOOTP server database.

Because of this time dependency, DHCP is also used to dynamically assign available IP addresses through client leases (lease time after which the client requests a new address) where each DHCP client address is saved temporarily in the server database.

In addition, DHCP clients do not require a system restart to rebind or renew configuration with the DHCP server. Instead, clients automatically enter a rebinding state at set timed intervals to renew their leased address allocation with the DHCP server. This process occurs in the background and is transparent to the user.
There are three different operating modes for a DHCP server:

- **Manual assignment**
  In this mode, the IP addresses are permanently assigned on the DHCP server to specific MAC addresses. The addresses are assigned to the MAC address for an indefinite period.
  Manual assignments are used primarily to ensure that the DHCP client can be reached under a fixed IP address.

- **Automatic assignment**
  For automatic assignment, a range of IP addresses is assigned on the DHCP server.
  If the address was assigned from this range once to a DHCP client, then it belongs to the client for an indefinite period as the assigned IP address is also bound to the MAC address.

- **Dynamic assignment**
  This process is similar to automatic assignment, but the DHCP server has a statement in its configuration file that specifies how long a certain IP address may be "leased" to a client before the client must log into the server again and request an "extension".
  If the client does not log in, the address is released and can be reassigned to another (or the same) client. The time defined by the administrator is called Lease Time.
  Some DHCP servers also assign IP addresses based on the MAC address, i.e., a client receives the same IP address as before after longer network absence and elapse of the Lease Time (unless the IP address has been assigned otherwise in the mean time).

DHCP is used to dynamically configure the network parameters.
The ETHERNET TCP/IP fieldbus controller has a DHCP client that supports the following options in addition to the default "IP address" option:

Table 75: Meaning of DHCP options

Option	Meaning
[OPT1] Subnet mask	32-bit address mask that displays which bits of the IP address identify the network and which identify the network stations.
[OPT2] Time zone	Time difference between the local time and the UTC (Universal Time Coordinated).
[OPT3] Gateway	IP address of the router that permits access to other networks.
[OPT6] DNS server	IP address of the name servers that converts a name into an IP address. Up to 2 DNS servers can be configured.
[OPT15] Domain name *)	The name of the domain is the unique name of a network. The domain name can contain up to 32 characters.
[OPT42] NTP server	IP address of the Network Time Server. When assigning an NTP server, the SNTP client is automatically enabled in the coupler.
[OPT51] Lease time	The maximum duration (i.e., how long the fieldbus coupler/controller maintains the assigned IP address) can be defined here. The maximum lease time for the fieldbus controller is 48 days. This is due to the internal timer resolution.
[OPT58] Renewing time	The renewing time indicates when the fieldbus coupler/controller must renew the lease time. The renewing time should be approximately half of the lease time.
[OPT59] Rebinding time	The rebinding time indicates after what amount of time the fieldbus coupler/controller must have received its new address. The rebinding time should be approximately 7/8 of the lease time.

*) In contrast to BootP, the DHCP client does not support assignment of the host name.

12.1.2.3 HTTP (Hypertext Transfer Protocol)

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

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

The HTTP server implemented in the (programmable) fieldbus coupler or controller is used for viewing the HTML pages saved in the coupler/controller. The HTML pages provide information about the coupler/controller (state, configuration), the network and the process image.

On some HTML pages, (programmable) fieldbus coupler or controller settings can also be defined and altered via the web-based management system (e.g. whether IP configuration of the coupler/controller is to be performed via the DHCP protocol, the BootP protocol or from the data stored in the EEPROM).

The HTTP server uses port number 80.

12.1.2.4 DNS (Domain Name Systems)

The DNS client enables conversion of logical Internet names such as www.wago.com into the appropriate decimal IP address represented with separator stops, via a DNS server. Reverse conversion is also possible.

The addresses of the DNS server are configured via DHCP, BootP or web-based management. Up to 2 DNS servers can be specified. The host identification can be achieved with two functions, an internal host table is not supported.

12.1.2.5 SNTP-Client (Simple Network Time Protocol)

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

Configuration of the SNTP client

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

Table 76: Meaning of the SNTP Parameters

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

12.1.2.6 FTP-Server (File Transfer Protocol)

The file transfer protocol (FTP) enables files to be exchanged between different network stations regardless of operating system.

In the case of the ETHERNET coupler/controller, FTP is used to store and read the HTML pages created by the user, the IEC61131 program and the IEC61131 source code in the (programmable) fieldbus coupler or controller.

A total memory of 2 MB is available for the file system.

Note

Cycles for flash limited to 1 million!
Up to 1 million write cycles per sector are allowed when writing the flash for the file system. The file system supports "Wear-Leveling", so that the same sectors are not always written to.

Information

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

12.1.2.7 SNMP (Simple Network Management Protocol)

The Simple Network Management Protocol (SNMP) is responsible for transporting the control data that allows the exchange of management information as well as status and statistic data between individual network components and a management system.

An SNMP management workstation polls the SNMP agents to obtain information on the relevant devices.

SNMP is supported in versions 1/2c and some fieldbus couplers/controllers in version 3.

This represents a community message exchange in SNMP versions 1 and 2c. The community name of the network community must thereby be specified.

In SNMP version 3, exchanging messages is user-related. Each device, that knows the passwords set via WBM, may read or write values from the controller. In SNMPv3, user data from SNMP messages can also be transmitted in encoded form. This way, both requested values and values to be written cannot be easily decoded by others via ETHERNET. This is why SNMPv3 is often used in safety-related networks.

The device data, that can be accessed or modified by an SNMP agent, is called SNMP object. The sets of SNMP objects are stored in a logical database called Management Information Base (MIB); this is why these objects are typically known as "MIB objects".

The SNMP of the ETHERNET controller includes both the general MIB acc. to RFC1213 (MIB II) and a special WAGO MIB.

SNMP is processed via port 161. The port number for SNMP traps (agent messages) is 161. Both ports must be enabled to use SNMP.

12.1.2.7.1 MIB II Description

MIB II acc. to RFC1213 is divided into the following groups:

Table 77: MIB II groups

Group	Identifier
System Group	1.3.6.1.2.1.1
Interface Group	1.3.6.1.2.1.2
IP Group	1.3.6.1.2.1.4
IpRoute Table Group	1.3.6.1.2.1.4.21
ICMP Group	1.3.6.1.2.1.5
TCP Group	1.3.6.1.2.1.6
UDP Group	1.3.6.1.2.1.7
SNMP Group	1.3.6.1.2.1.11

Information

Additional Information:
Please find detailed information on these individual groups in section "MIB II groups" of the manual appendix..

12.1.2.7.2 Traps

Standard Traps

For specific events, the SNMP agent will independently send one of the following messages without polling the manager.

Note

Enable event messages (traps) in the WBM!
Initially enable the event messages in the WBM in menu "SNMP“ under "Trap Enable“. Traps in version 1, 2c and 3 may be activated separately.

The following messages are triggered automatically as traps (SNMPv1) by the fieldbus coupler/controller:

Table 78: Standard Traps

TrapType/TrapNumber/ OID of the provided value	Name	Event
TrapType = 0	ColdStart	Restart the coupler/controller
TrapType = 1	WarmStart	Reset via service switch
TrapType = 3	EthernetUp	Network connection detected
TrapType = 4	AuthenticationFailure	Unauthorized (abortive) MIB access
TrapType = 6/ ab Trap-Nummer 25 benutzerspezifisch	enterpriseSpecific	Enterprise-specific messages and function poll in the PFC program starting with enterprise trap number 25

12.1.3 Application Protocols

If fieldbus specific application protocols are implemented, then the appropriate fieldbus specific communication is possible with the respective coupler/controller. Thus the user is able to have a simple access from the respective fieldbus on the fieldbus node.

The implemented fieldbus specific application protocols these protocols are individual described in the following chapters.

12.2 MODBUS Functions

12.2.1 General

MODBUS is a manufacturer-independent, open fieldbus standard for diverse applications in manufacturing and process automation.

The MODBUS protocol is implemented according to the current Internet Draft of the IETF (Internet Engineering Task Force) and performs the following functions:

- Transmission of the process image
- Transmission of the fieldbus variables
- Transmission of different settings and information on the coupler/controller

The data transmission in the fieldside takes place via TCP and via UDP.

The MODBUS/TCP protocol is a variation of the MODBUS protocol, which was optimized for communication via TCP/IP connections.

This protocol was designed for data exchange in the field level (i.e. for the exchange of I/O data in the process image).

All data packets are sent via a TCP connection with the port number 502.

MODBUS/TCP segment

The general MODBUS/TCP header is as follows:

Table 79: MODBUS/TCP header

Byte	0	1	2	3	4	5	6	7	8…n
	Identifier (entered by receiver)		Protocol-identifier (is always 0)		Length field (High byte, low byte)		Unit identifier (Slave address)	MODBUS function code	Data

Information

Additional Information
The structure of a datagram is specific for the individual function. Refer to the descriptions of the MODBUS Function codes.

For the MODBUS protocol 15 connections are made available over TCP. Thus it allows digital and analog output data to be directly read out at a fieldbus node and special functions to be executed by way of simple MODBUS function codes from 15 stations simultaneously.

For this purpose a set of MODBUS functions from the Open MODBUS/TCP specification is realized.

Information

More information
More information on the "Open MODBUS/TCP specification" you can find in the Internet: www.modbus.org .

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

Table 80: Basic data types of MODBUS protocol

Data type	Length	Description
Discrete Inputs	1 Bit	Digital inputs
Coils	1 Bit	Digital outputs
Input Register	16 Bit	Analog input data
Holding Register	16 Bit	Analog output data

For each basic data type one or more function codes are defined.

These functions allow digital or analog input and output data, and internal variables to be set or directly read out of the fieldbus node.

Table 81: List of the MODBUS functions in the fieldbus controller

Function code		Function	Access method and description	Access to resources	
FC1	0x01	Read Coils	Reading of several single input bits	**R:**	Process image, PFC variables
FC2	0x02	Read Input Discretes	Reading of several input bits	**R:**	Process image, PFC variables
FC3	0x03	Read Multiple Registers	Reading of several input registers	**R:**	Process image, PFC variables, internal variables, NOVRAM
FC4	0x04	Read Input Registers	Reading of several input registers	**R:**	Process image, PFC variables, internal variables, NOVRAM
FC5	0x05	Write Coil	Writing of an individual output bit	**W:**	Process image, PFC variables
FC6	0x06	Write Single Register	Writing of an individual output register	**W:**	Process image, PFC variables, internal variables, NOVRAM
FC11	0x0B	Get Comm Event Counters	Communication event counter	**R:**	None
FC15	0x0F	Force Multiple Coils	Writing of several output bits	**W:**	Process image, PFC variables
FC16	0x10	Write Multiple Registers	Writing of several output registers	**W:**	Process image, PFC variables, internal variables, NOVRAM
FC22	0x16	Mask Write Register		**W:**	Process image, PFC variables, NOVRAM
FC23	0x17	Read/Write Registers	Reading and writing of several output registers	**R/W:**	Process image, PFC variables, NOVRAM

To execute a desired function, specify the respective function code and the address of the selected input or output data.

Note

Note the number system when addressing!
The examples listed use the hexadecimal system (i.e.: 0x000) as their numerical format. Addressing begins with 0. The format and beginning of the addressing may vary according to the software and the control system. All addresses then need to be converted accordingly.

12.2.2 Use of the MODBUS Functions

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

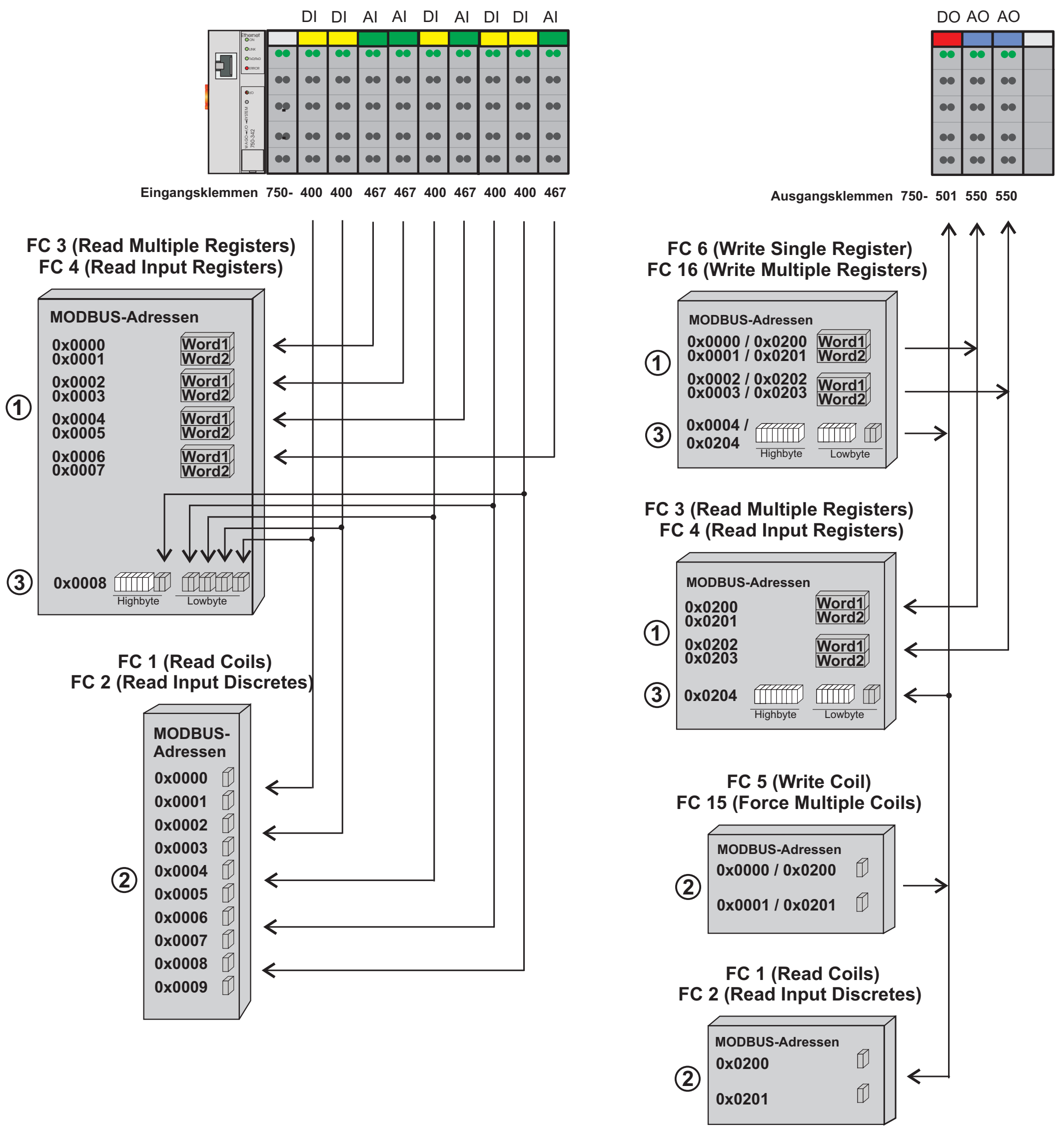


Figure 76: Use of the MODBUS Functions

Note

Use register functions to access analog signals and coil functions to access binary signals!

It is recommended that analog data be accessed with register functions ① and digital data with coil functions ②. If reading or writing access to binary signals is performed via register functions ③, an address shift may occur as soon as further analog modules are operated on the coupler/controller.

12.2.3 Description of the MODBUS Functions

All MODBUS functions are executed as follows:

1. A MODBUS TCP master (e.g., a PC) makes a request to the WAGO fieldbus node using a specific function code based on the desired operation..

2. The WAGO fieldbus node receives the datagram and then responds to the master with the proper data, which is based on the master's request.

If the WAGO fieldbus node receives an incorrect request, it sends an error datagram (Exception) to the master.
The exception code contained in the exception has the following meaning:

Table 82: Exception Codes

Exception Code	Meaning
0x01	Illegal function
0x02	Illegal data address
0x03	Illegal data value
0x04	Slave device failure

The following chapters describe the datagram architecture of request, response and exception with examples for each function code.

Note

Reading and writing of outputs via FC1 to FC4 is also possible by adding an offset!

In the case of the read functions (FC1 ... FC4) the outputs can be additionally written and read back by adding an offset of 200hex (0x0200) to the MODBUS addresses in the range of [0_{hex} ... FF_{hex}] and an offset of 1000_{hex} (0x01000) to the MODBUS addresses in the range of [6000_{hex} ... $62FC_{hex}$].

12.2.3.1 Function Code FC1 (Read Coils)

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

Request

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

Table 83: Request of Function code FC1

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x01
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0008

Response

The current values of the response bits are packed in the data field. A binary 1 corresponds to the ON status and a 0 to the OFF status. The lowest value bit of the first data byte contains the first bit of the request. The others follow in ascending order. If the number of inputs is not a multiple of 8, the remaining bits of the last data byte are filled with zeroes (truncated).

Table 84: Response of Function code FC1

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x01
Byte 8	Byte count	0x01
Byte 9	Bit values	0x12

The status of the inputs 7 to 0 is shown as byte value 0x12 or binary 0001 0010. Input 7 is the bit having the highest significance of this byte and input 0 the lowest value.
The assignment is thus made from 7 to 0 as follows:

Table 85: Assignment of inputs

	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF
Bit	0	0	0	1	0	0	1	0
Coil	7	6	5	4	3	2	1	0

Exception

Table 86: Exception of Function code FC1

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x81
Byte 8	Exception code	0x01 or 0x02

12.2.3.2 Function Code FC2 (Read Input Discretes)

This function reads the input bits from a slave device.

Request

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

Table 87: Request of Function code FC2

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x02
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0008

Response

The current value of the requested bits are packed into the data field. A binary 1 corresponds to the ON status and a 0 the OFF status. The lowest value bit of the first data byte contains the first bit of the inquiry. The others follow in an ascending order. If the number of inputs is not a multiple of 8, the remaining bits of the last data byte are filled with zeroes (truncated).

Table 88: Response of Function code FC2

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x02
Byte 8	Byte count	0x01
Byte 9	Bit values	0x12

The status of the inputs 7 to 0 is shown as a byte value 0x12 or binary 0001 0010. Input 7 is the bit having the highest significance of this byte and input 0 the lowest value. The assignment is thus made from 7 to 0 as follows:

Table 89: Assignment of inputs

	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF
Bit	0	0	0	1	0	0	1	0
Coil	7	6	5	4	3	2	1	0

Exception

Table 90: Exception of Function code FC2

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x82
Byte 8	Exception code	0x01 or 0x02

12.2.3.3 Function Code FC3 (Read Multiple Registers)

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

Request

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

Table 91: Request of Function code FC3

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x03
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002

Response

The reply register data is packed as 2 bytes per register. The first byte contains the higher value bits, the second the lower values.

Table 92: Response of Function code FC3

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x03
Byte 8	Byte count	0x04
Byte 9, 10	Value register 0	0x1234
Byte 11, 12	Value register 1	0x2345

The contents of register 0 are displayed by the value 0x1234 and the contents of register 1 is 0x2345.

Exception

Table 93: Exception of Function code FC3

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x83
Byte 8	Exception code	0x01 or 0x02

12.2.3.4 Function Code FC4 (Read Input Registers)

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

Request

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

Table 94: Request of Function code FC4

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x04
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002

Response

The register data of the response is packed as 2 bytes per register. The first byte has the higher value bits, the second the lower values.

Table 95: Response of Function code FC4

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x04
Byte 8	Byte count	0x04
Byte 9, 10	Value register 0	0x1234
Byte 11, 12	Value register 1	0x2345

The contents of register 0 are shown by the value 0x1234 and the contents of register 1 is 0x2345.

Exception

Table 96: Exception of Function code FC4

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x84
Byte 8	Exception code	0x01 or 0x02

12.2.3.5 Function Code FC5 (Write Coil)

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

Request

The request specifies the reference number (output address) of output bit to be written. The reference number of the request is zero based; therefore, the first coil starts at address 0.
Example: Turn ON the second output bit (address 1)

Table 97: Request of Function code FC5

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x05
Byte 8, 9	Reference number	0x0001
Byte 10	ON/OFF	0xFF
Byte 11		0x00

Response

Table 98: Response of Function code FC5

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x05
Byte 8, 9	Reference number	0x0001
Byte 10	Value	0xFF
Byte 11		0x00

Exception

Table 99: Exception of Function code FC5

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01, 0x02 or 0x03

12.2.3.6 Function Code FC6 (Write Single Register)

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

Request

The request specifies the reference number (register address) of the first output word to be written. The value to be written is specified in the "Register Value" field. The reference number of the request is zero based; therefore, the first register starts at address 0.
Example: Write a value of 0x1234 to the second output register

Table 100: Request of Function code FC6

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x06
Byte 8, 9	Reference number	0x0001
Byte 10, 11	Register value	0x1234

Response

The reply is an echo of the inquiry.

Table 101: Response of Function code FC6

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x06
Byte 8, 9	Reference number	0x0001
Byte 10, 11	Register value	0x1234

Exception

Table 102: Exception of Function code FC6

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.7 Function Code FC11 (Get Comm Event Counter)

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

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

Request

Table 103: Request of Function code FC11

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0002
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x0B

Response

The reply contains a 2-byte status word and a 2-byte event counter. The status word only contains zeroes.

Table 104: Response of Function code FC11

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x0B
Byte 8, 9	Status	0x0000
Byte 10, 11	Event count	0x0003

The event counter shows that 3 (0x0003) events were counted.

Exception

Table 105: Exception of Function code FC 11

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.8 Function Code FC15 (Force Multiple Coils)

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

Request

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

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

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

Table 106: Request of Function code FC15

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0009
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x0F
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0010
Byte 12	Byte count	0x02
Byte 13	Data byte1	0xA5
Byte 14	Data byte2	0xF0

Response

Table 107: Response of Function code FC15

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x0F
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit count	0x0010

Exception

Table 108: Exception of Function code FC15

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x8F
Byte 8	Exception code	0x01 or 0x02

12.2.3.9 Function Code FC16 (Write Multiple Registers)

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

Request

The Request specifies the reference number (starting register), the word count (number of registers to write), and the register data . The data is sent as 2 bytes per register. The registers are zero-based; therefore, the first output is at address 0. Example: Set data in registers 0 and 1

Table 109: Request of Function code FC16

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x000B
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002
Byte 12	Byte count	0x04
Byte 13, 14	Register value 1	0x1234
Byte 15, 16	Register value 2	0x2345

Response

Table 110: Response of Function code FC16

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word count	0x0002

Exception

Table 111: Exception of Function code FC16

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.10 Function Code FC22 (Mask Write Register)

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

Request

Table 112: Request of Function code FC22

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x0002
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x16
Byte 8, 9	Reference number	0x0000
Byte 10, 11	AND mask	0x0000
Byte 12, 13	OR mask	0xAAAA

Response

Table 113: Response of Function code FC22

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	AND mask	0x0000
Byte 12, 13	OR mask	0xAAAA

Exception

Table 114: Exception of Function code FC22

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

12.2.3.11 Function Code FC23 (Read/Write Multiple Registers)

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

Request

The reference numbers (addresses) are zero-based in the request message; therefore, the first register is at address 0.
The request message specifies the registers to read and write. The data is sent as 2 bytes per register.
Example: The data in register 3 is set to value 0x0123, and values 0x0004 and 0x5678 are read out of the two registers 0 and 1.

Table 115: Request of Function code FC23

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	Protocol identifier	0x0000
Byte 4, 5	Length field	0x000F
Byte 6	Unit identifier	0x01 not used
Byte 7	MODBUS function code	0x17
Byte 8, 9	Reference number for read	0x0000
Byte 10, 11	Word count for read (1…125)	0x0002
Byte 12, 13	Reference number for write	0x0003
Byte 14, 15	Word count for write (1…100)	0x0001
Byte 16	Byte count (2 x word count for write)	0x02
Byte 17...(B+16)	Register values (B = Byte count)	0x0123

Response

Table 116: Response of Function code FC23

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x17
Byte 8	Byte count (2 x word count for read)	0x04
Byte 9...(B+8)	Register values (B = Byte count)	0x0004 or 0x5678

Exception

Table 117: Exception of Function code FC23

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x97
Byte 8	Exception code	0x01 or 0x02

Note

Note that if the register ranges overlap, the results are undefined!
If register areas for read and write overlap, the results are undefined.

12.2.4 MODBUS Register Mapping

The following tables display the MODBUS addressing and the corresponding IEC61131 addressing for the process image, the PFC variables, the 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 Access Reading (with FC3, FC4 and FC23)

Table 118: Register access reading (with FC3, FC4 and FC23)

MODBUS address		IEC 61131 address	Memory range
[dec]	[hex]		
0...255	0x0000...0x00FF	%IW0...%IW255	Physical input area (1) First 256 words of physical input data
256...511	0x0100...0x01FF	%QW256...%QW511	PFC OUT area Volatile PFC output variables
512...767	0x0200...0x02FF	%QW0...%QW255	Physical output area (1) First 256 words of physical output data
768...1023	0x0300...0x03FF	%IW256...%IW511	PFC IN area Volatile PFC input variables
1024...4095	0x0400...0x0FFF	-	MODBUS exception: “Illegal data address”
4096...12287	0x1000...0x2FFF	-	Configuration register (see following chapter “Configuration Functions“)
12288...24575	0x3000...0x5FFF	%MW0...%MW12287	NOVRAM 24 kB retain memory *) *) In Target settings RETAIN on 0, flags on MAX (24 kB)
24576...25340	0x6000...0x62FC	%IW512...%IW1275	Physical input area (2) Additional 764 words physical input data
25341...28671	0x62FD...0x6FFF	-	MODBUS exception: “Illegal data address”
28672...29435	0x7000...0x72FB	%QW512...%QW1275	Physical output area (2) Additional 764 words physical output data
29436...32767	0x72FC...0x7FFF	-	MODBUS exception: “Illegal data address”
32768...36863	0x8000...0x8FFF	-	NOVRAM Additional, 16-bit length
36864...65535	0x9000...0xFFFF	-	MODBUS exception: “Illegal data address”

Register Access Writing (with FC6, FC16, FC22 and FC23)

Table 119: Register access writing (with FC6, FC16, FC22 and FC23)

MODBUS address [dec]	[hex]	IEC 61131 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 „Configuration Functions“)
12288...24575	0x3000...0x5FFF	%MW0...%MW12287	NOVRAM 24 kB retain memory *) *) In Target settings RETAIN on 0, flags on MAX (24 kB)
24576...25340	0x6000...0x62FC	%QW512...%QW1275	Physical output area (2) Additional 764 words physical output data
25341...28671	0x62FD...0x6FFF	-	MODBUS exception: “Illegal data address”
28672...29435	0x7000...0x72FB	%QW512...%QW1275	Physical output area (2) Additional 764 words physical output data
29436...32767	0x72FC...0x7FFF	-	MODBUS exception: “Illegal data address”
32768...36863	0x8000...0x8FFF	-	NOVRAM Additional, 16-bit length
36864...65535	0x9000...0xFFFF	-	MODBUS exception: “Illegal data address”

The digital MODBUS services (coil services) are bit accesses, with which only the states of digital I/O modules can be determined or changed. Complex I/O modules are not attainable with these services and so they are ignored. Because of this the addressing of the digital channels begins again with 0, so that the MODBUS address is always identical to the channel number, (i.e. the digital input no. 47 has the MODBUS address "46").

Bit Access Reading (with FC1 and FC2)

Table 120: Bit access reading (with FC1 and FC2)

MODBUS address		Memory range	Description
[dec]	[hex]		
0...511	0x0000...0x01FF	Physical input area (1)	First 512 digital inputs
512...1023	0x0200...0x03FF	Physical output area (1)	First 512 digital outputs
1024...4095	0x0400...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...8191	0x1000...0x1FFF	%QX256.0...%QX511.15	PFC OUT area Volatile PFC output variables
8192...12287	0x2000...0x2FFF	%IX256.0...%IX511.15	PFC IN area Volatile PFC input variables
12288...32767	0x3000...0x7FFF	%MX0...%MX1279.15	NOVRAM 2 kB retain memory (max. 24 kB)
32768...34295	0x8000...0x85F7	Physical input area (2)	Starts with the 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)

Table 121: Bit access writing (with FC5 and FC15)

MODBUS address		Memory Range	Description
[dez]	[hex]		
0...511	0x0000...0x01FF	Physical output area (1)	First 512 digital outputs
512...1023	0x0200...0x03FF	Physical output area (1)	First 512 digital outputs
1024...4095	0x0400...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...8191	0x1000...0x1FFF	%IX256.0...%IX511.15	PFC IN area Volatile PFC input variables
8192...12287	0x2000...0x2FFF	%IX256.0...%IX511.15	PFC IN area Volatile PFC input variables
12288...32767	0x3000...0x7FFF	%MX0...%MX1279.15	NOVRAM 2 kB retain memory
32768...34295	0x8000...0x85F7	Physical output area (2)	Starts with the 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"

12.2.5 MODBUS Registers

Table 122: MODBUS registers

Register address	Access	Length (Word)	Description
0x1000	R/W	1	Watchdog time read/write
0x1001	R/W	1	Watchdog coding mask 1…16
0x1002	R/W	1	Watchdog coding mask 17…32
0x1003	R/W	1	Watchdog trigger
0x1004	R	1	Minimum trigger time
0x1005	R/W	1	Watchdog stop (Write sequence 0xAAAA, 0x5555)
0x1006	R	1	Watchdog status
0x1007	R/W	1	Restart watchdog (Write sequence 0x1)
0x1008	R/W	1	Stop watchdog (Write sequence 0x55AA or 0xAA55)
0x1009	R/W	1	MODBUS and HTTP close at watchdog time-out
0x100A	R/W	1	Watchdog configuration
0x100B	W	1	Save watchdog parameter
0x1020	R	1…2	LED error code
0x1021	R	1	LED error argument
0x1022	R	1…4	Number of analog output data in the process image (in bits)
0x1023	R	1…3	Number of analog input data in the process image (in bits)
0x1024	R	1…2	Number of digital output data in the process image (in bits)
0x1025	R	1…4	Number of digital input data in the process image (in bits)
0x1028	R/W	1	Boot configuration
0x1029	R	9	MODBUS/TCP statistics
0x102A	R	1	Number of TCP connections
0x102B	W	1	KBUS Reset
0x1030	R/W	1	Configuration MODBUS/TCP time-out
0x1031	R	3	Read out the MAC-ID of the coupler/controller
0x1035	R/W	1	Timeoffset RTC
0x1036	R/W	1	Daylight Saving
0x1037	R/W	1	Modbus Response Delay (ms)
0x1050	R	3	Diagnosis of the connected I/O modules
0x2000	R	1	Constant 0x0000
0x2001	R	1	Constant 0xFFFF
0x2002	R	1	Constant 0x1234
0x2003	R	1	Constant 0xAAAA
0x2004	R	1	Constant 0x5555
0x2005	R	1	Constant 0x7FFF
0x2006	R	1	Constant 0x8000
0x2007	R	1	Constant 0x3FFF
0x2008	R	1	Constant 0x4000
0x2010	R	1	Firmware version
0x2011	R	1	Series code
0x2012	R	1	Coupler/controller code
0x2013	R	1	Firmware version major revision
0x2014	R	1	Firmware version minor revision

Table 123: MODBUS registers (Continuation)

Register address	Access	Length (Word)	Description
0x2020	R	16	Short description controller
0x2021	R	8	Compile time of the firmware
0x2022	R	8	Compile date of the firmware
0x2023	R	32	Indication of the firmware loader
0x2030	R	65	Description of the connected I/O modules (module 0…64)
0x2031	R	64	Description of the connected I/O modules (module 65…128)
0x2032	R	64	Description of the connected I/O modules (module 129…192)
0x2033	R	63	Description of the connected I/O modules (module 193…255)
0x2040	W	1	Software reset (Write sequence 0x55AA or 0xAA55)
0x2041	W	1	Format flash disk
0x2042	W	1	Extract HTML sides from the firmware
0x2043	W	1	Factory settings

12.2.5.1 Accessing Register Values

You can use any MODBUS application to access (read from or write to) register values. Both commercial (e.g., "Modscan") and free programs (from http://www.modbus.org/tech.php) are available.

The following sections describe how to access both the registers and their values.

12.2.5.2 Watchdog Registers

The watchdog monitors the data transfer between the fieldbus master and the controller. Every time the controller receives a specific request (as define in the watchdog setup registers) from the master, the watchdog timer in the controller resets.

In the case of fault free communication, the watchdog timer does not reach its end value. After each successful data transfer, the timer is reset.

If the watchdog times out, a fieldbus failure has occurred. In this case, the fieldbus controller answers all following MODBUS TCP/IP requests with the exception code 0x0004 (Slave Device Failure).

In the controller special registers are used to setup the watchdog by the master (Register addresses 0x1000 to 0x1008).

By default, the watchdog is not enabled when you turn the controller on. To activate it, the first step is to set/verify the desired time-out value of the Watchdog Time register (0x1000). Second, the function code mask must be specified in the mask register (0x1001), which defines the function code(s) that will reset the timer for the first time. Finally, the Watchdog-Trigger register (0x1003) or the register 0x1007 must be changed to a non-zero value to start the timer subsequently.

Reading the Minimum Trigger time (Register 0x1004) reveals whether a watchdog fault occurred. If this time value is 0, a fieldbus failure is assumed. The timer of watchdog can manually be reset, if it is not timed out, by writing a value of 0x1 to the register 0x1003 or to the Restart Watchdog register 0x1007.

After the watchdog is started, it can be stopped by the user via the Watchdog Stop register (0x1005) or the Simply Stop Watchdog register (0x1008).

The watchdog registers can be addressed in the same way as described with the MODBUS read and write function codes. Specify the respective register address in place of the reference number.

Table 124: Register address 0x1000

Register address 0x1000 (4096$_{dec}$)	
Value	Watchdog time, WS_TIME
Access	Read/write
Default	0x0064
Description	This register stores the watchdog timeout value as an unsigned 16 bit value. The default value is 0. Setting this value will not trigger the watchdog. However, a non zero value must be stored in this register before the watchdog can be triggered. The time value is stored in multiples of 100ms (e.g., 0x0009 is .9 seconds). It is not possible to modify this value while the watchdog is running.

Table 125: Register address 0x1001

Register address 0x1001 (4097$_{dec}$)	
Value	Watchdog function coding mask, function code 1...16, WDFCM_1_16
Access	Read/write
Default	0xFFFF
Description	Using this mask, the function codes can be set to trigger the watchdog function. The function code can be selected via a "1" FC 1 Bit 0 FC 2 Bit 1 FC 3 Bit 2 FC 4 Bit 3 FC 5 Bit 4 ... FC 16 Bit 15 Changes to the register value can only be made if the watchdog is deactivated. The bit pattern stored in the register defines the function codes that trigger the watchdog. Some function codes are not supported. For those the watchdog will not be triggered even if another MODBUS device transmits one of them.

Table 126: Register address 0x1002

Register address 0x1002 (4098$_{dec}$)	
Value	Watchdog function coding mask, function code 17...32, WD_FCM_17_32
Access	Read/write
Default	0xFFFF
Description	Same function as above, however, with the function codes 17 to 32. FC 17 Bit 0 FC 18 Bit 1 ... FC 32 Bit 15 These codes are currently not supported, for this reason the default value should not be changed. Changes to the register value can only be made if the watchdog is deactivated. It is not possible to modify this value while the watchdog is running.

Table 127: Register address 0x1003

Register address 0x1003 (4099$_{dez}$)	
Value	Watchdog Trigger, WD_TRIGGER
Access	Read/write
Standard	0x0000
Description	This register is used for an alternative trigger method. The watchdog is triggered by writing different values in this register. Values following each other must differ in size. Writing of a value not equal to zero starts the watchdog after a Power-on. For a restart the written value must necessarily be unequal the before written value! A watchdog fault is reset and writing process data is possible again.

Table 128: Register address 0x1004

Register address 0x1004 (4100$_{dez}$)	
Value	Minimum current trigger time, WD_AC_TRG_TIME
Access	Read/write
Standard	0xFFFF
Description	This register saves the minimum current watchdog trigger time. If the watchdog is triggered, the saved value is compared with the current value. If the current value is smaller than the saved value, this is replaced by the current value. The unit is 100 ms/digit. The saved value is changed by writing new values, which does not affect the watchdog. 0x000 is not permissible.

Table 129: Register address 0x1005

Register address 0x1005 (4101$_{dez}$)	
Value	Stop watchdog, WD_AC_STOP_MASK
Access	Read/write
Standard	0x0000
Description	The watchdog is stopped if here the value 0xAAAA is written first, followed by 0x5555. The watchdog fault reaction is blocked. A watchdog fault is reset and writing on the process data is possible again.

Table 130: Register address 0x1006

Register address 0x1006 (4102$_{dez}$)	
Value	While watchdog is running, WD_RUNNING
Access	Read
Standard	0x0000
Description	Current watchdog status. at 0x0000: Watchdog not active at 0x0001: Watchdog active at 0x0002: Watchdog exhausted.

Table 131: Register address 0x1007

Register address 0x1007 (4103$_{dez}$)	
Value	Restart watchdog, WD_RESTART
Access	Read/write
Standard	0x0001
Description	This register restarts the watchdog timer by writing a value of 0x1 into it. If the watchdog was stopped before the overrun, it is not restarted.

Table 132: Register address 0x1008

Register address 0x1008 (4104$_{dez}$)	
Value	Simply stop watchdog, WD_AC_STOP_SIMPLE
Access	Read/write
Standard	0x0000
Description	This register stops the watchdog by writing the value 0x0AA55 or 0X55AA into it. The watchdog timeout fault is deactivated and it is possible to write in the watchdog register again. If there is an existing watchdog fault, it is reset

Table 133: Register address 0x1009

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

Table 134: Register address 0x100A

Register address 0x100A (4106$_{dez}$)	
Value	Alternative watchdog
Access	Read/write
Standard	0x0000
Description	This register provides an alternate way to activate the watchdog timer. Procedure: Write a time value in register 0x1000; then write a 0x0001 into register 0x100A. With the first MODBUS request, the watchdog is started. The watchdog timer is reset with each MODBUS/TCP instruction. If the watchdog times out, all outputs are set to zero. The outputs will become operational again, after communications are re-established. Register 0x00A is non-volatile, including register 0x1000. It is not possible to modify the time value in register 0x1000 while the watchdog is running.

The length of each register is 1 word; i.e., with each access only one word can be written or read. Following are two examples of how to set the value for a time overrun:

Setting the watchdog for a timeout of more than 1 second:

1. Write 0x000A in the register for time overrun (0x1000).
   Register 0x1000 works with a multiple of 100 ms;
   1 s = 1000 ms; 1000 ms / 100 ms = 10_{dec} = A_{hex})

2. Use the function code 5 to write 0x0010 (=2(5-1)) in the coding mask (register 0x1001).

Table 135: Starting Watchdog

FC	FC16	FC15	FC14	FC13	FC12	FC11	FC10	FC9	FC8	FC7	FC6	**FC5**	FC4	FC3	FC2	FC1
Bit	15	14	13	12	11	10	9	8	7	6	5	**4**	3	2	1	0
bin	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0
hex	0				0				**1**				0			

Function code 5 (writing a digital output bit) continuously triggers the watchdog to restart the watchdog timer again and again within the specified time. If time between requests exceeds 1 second, a watchdog timeout error occurs.

3. To stop the watchdog, write the value 0x0AA55 or 0X55AA into 0x1008 (Simply Stop Watchdog register, WD_AC_STOP_SIMPLE).

Setting the watchdog for a timeout of 10 minutes or more:

1. Write 0x1770 (= 10*60*1000 ms / 100 ms) in the register for time overrun (0x1000).
   (Register 0x1000 works with a multiple of 100 ms;
   10 min = 600,000 ms; 600,000 ms / 100 ms = 6000dec = 1770hex)

2. Write 0x0001 in the watchdog trigger register (0x1003) to start the watchdog.

3. Write different values (e.g., counter values 0x0000, 0x0001) in the watchdog to trigger register (0x1003).

Values following each other must differ in size. Writing of a value not equal to zero starts the watchdog. Watchdog faults are reset and writing process data is possible again.

4. To stop the watchdog, write the value 0x0AA55 or 0X55AA into 0x1008 (Simply Stop Watchdog register, WD_AC_STOP_SIMPLE).

Table 136: Register address 0x100B

Register address 0x100B (4107$_{dez}$)	
Value	Save watchdog parameter
Access	Write
Standard	0x0000
Description	With writing of '0x55AA' or '0xAA55' in register 0x100B the registers 0x1000, 0x1001, 0x1002 are set on remanent.

12.2.5.3 Diagnostic Registers

The following registers can be read to determine errors in the node:

Table 137: Register address 0x1020

Register address 0x1020 (4128$_{dec}$)	
Value	LedErrCode
Access	Read
Description	Declaration of the Error code

Table 138: Register address 0x1021

Register address 0x1021 (4129$_{dec}$)	
Value	LedErrArg
Access	Read
Description	Declaration of the Error argument

12.2.5.4 Configuration Registers

The following registers contain configuration information of the connected modules:

Table 139: Register address 0x1022

Register address 0x1022 (4130$_{dec}$)	
Value	CnfLen.AnalogOut
Access	Read
Description	Number of word-based outputs registers in the process image in bits (divide by 16 to get the total number of analog words)

Table 140: Register address 0x1023

Register address 0x1023 (4131$_{dec}$)	
Value	CnfLen.AnalogInp
Access	Read
Description	Number of word-based inputs registers in the process image in bits (divide by 16 to get the total number of analog words)

Table 141: Register address 0x1024

Register address 0x1024 (4132$_{dec}$)	
Value	CnfLen.DigitalOut
Access	Read
Description	Number of digital output bits in the process image

Table 142: Register address 0x1025

Register address 0x1025 (4133$_{dec}$)	
Value	CnfLen.DigitalInp
Access	Read
Description	Number of digital input bits in the process image

Table 143: Register address 0x1028

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

Table 144: Register address 0x1029

Register address 0x1029 (4137$_{dec}$) with 9 words		
Value	MODBUS TCP statistics	
Access	Read/write	
Description	1 word SlaveDeviceFailure	→ internal bus error, fieldbus error by activated watchdog
	1 word BadProtocol	→ error in the MODBUS TCP header
	1 word BadLength	→ Wrong telegram length
	1 word BadFunction	→ Invalid function code
	1 word BadAddress	→ Invalid register address
	1 word BadData	→ Invalid value
	1 word TooManyRegisters	→ Number of the registers which can be worked on is too large, Read/Write 125/100
	1 word TooManyBits	→ Number of the coils which can be worked on is too large, Read/Write 2000/800
	1 word ModTcpMessageCounter	→ Number of received MODBUS/TCP requests
	With Writing 0xAA55 or 0x55AA in the register will reset this data area.	

Table 145: Register address 0x102A

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

Table 146: Register address 0x102B

Register address 0x102B (4139$_{dez}$) with a word count of up to 1	
Value	KBUS reset
Access	Write
Description	Writing of this register restarts the internal bus

Table 147: Register address 0x1030

Register address 0x1030 (4144$_{dec}$) with a word count of 1	
Value	Configuration MODBUS/TCP Time-out
Access	Read/write
Default	0x0258 (600 decimal)
Description	This is the maximum number of milliseconds the fieldbus coupler will allow a MODBUS/TCP connection to stay open without receiving a MODBUS request. Upon time-out, idle connection will be closed. Outputs remain in last state. Default value is 600 ms (60 seconds), the time base is 100 ms, the minimal value is 100 ms. If the value is set to ‘0’, the timeout is disabled. On this connection, the watchdog is triggered with a request.

Table 148: Register address 0x1031

Register address 0x1031 (4145$_{dec}$) with a word count of 3	
Value	Read the MAC-ID of the controller
Access	Read
Description	This register gives the MAC-ID, with a length of 3 words

Table 1: Register address 0x1035

Register address 0x1035 (4149$_{dez}$) 1 Word	
Value	Configuration of the time offsets to the GMT time
Access	Read/write
Default	0x0000
Description	Register to set the time offset to the UTC time (Greenwich meridian) with a possible setting range from -12 to +12.

Table 1: Register address 0x1036

Register address 0x1036 (4150$_{dez}$) 1 Word	
Value	Configuration of summer or winter time
Access	Read/write
Default	0x0000
Description	Register to set winter or summer time (Daylight Saving Time). The values 0 and 1 are valid.

Table 1: Register address 0x1037

Register address 0x1031 (4151$_{dez}$) with a word count of 3	
Value	Configuration of Modbus Response Delay Time
Access	Read/write
Default	0x0000
Description	This register saves the value for the Modbus Response Delay Time for a Modbus connection. The time base is 1 ms. On the Modbus TCP connection, the response will be delayed by the inscribed time.

Table 149: Register address 0x1050

Register address 0x1050 (4176$_{dec}$) with a word count of 3	since Firmware version 9
Value	Diagnosis of the connected I/O modules
Access	Read
Description	Diagnosis of the connected I/O modules, length 3 words Word 1: Number of the module Word 2: Number of the channel Word 3: Diagnosis

Table 150: Register address 0x2030

Register address 0x2030 (8240$_{dec}$) with a word count of up to 65	
Value	Description of the connected I/O modules
Access	Read module 0...64
Description	Length 1...65 words These 65 registers identify the controller and the first 64 modules present in a node. Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 → Input module Bit position 1 → Output module Bit position 2…7 → Not used Bit position 8…14 → Module size in bits Bit position 15 → Designation digital module

Examples:

4 Channel Digital Input Module = 0x8401

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Code	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
Hex	8				4				0				1			

2 Channel Digital Output Module = 0x8202

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Code	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
Hex	8				2				0				2			

Table 151: Register address 0x2031

Register address 0x2031 (8241$_{dec}$) with a word count of up to 64	
Value	Description of the connected I/O modules
Access	Read modules 65...128
Description	Length 1-64 words These 64 registers identify the 2nd block of I/O modules present (modules 65 to 128). Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 → Input module Bit position 1 → Output module Bit position 2…7 → Not used Bit position 8…14 → Module size in bits Bit position 15 → Designation digital module

Table 152: Register address 0x2032

Register address 0x2032 (8242$_{dec}$) with a word count of up to 64	
Value	Description of the connected I/O modules
Access	Read modules 129...192
Description	Length 1…64 words These 64 registers identify the 3rd block of I/O modules present (modules 129 to 192). Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 → Input module Bit position 1 → Output module Bit position 2…7 → Not used Bit position 8…14 → Module size in bits Bit position 15 → Designation digital module

Table 153: Register address 0x2033

Register address 0x2033 (8243$_{dec}$) with a word count of up to 65	
Value	Description of the connected I/O modules
Access	Read modules 193 ... 255
Description	Length 1-63 words These 63 registers identify the 4th block of I/O modules present (modules 193 to 255). Each module is represented in a word. Because order numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 → Input module Bit position 1 → Output module Bit position 2…7 → Not used Bit position 8…14 → Module size in bits Bit position 15 → Designation digital module

Table 154: Register address 0x2040

Register address 0x2040 (8256$_{dec}$)	
Value	Implement a software reset
Access	Write (Write sequence 0xAA55 or 0x55AA)
Description	With Writing 0xAA55 or 0x55AA the register will be reset.

Table 155: Register address 0x2041

Register address 0x2041 (8257$_{dez}$)	
Value	Flash Format
Access	Write (Write sequence 0xAA55 or 0x55AA)
Description	The file system Flash is again formatted.

Table 156: Register address 0x2042

Register address 0x2042 (8258$_{dez}$)	
Value	Extract data files
Access	Write (Write sequence 0xAA55 or 0x55AA)
Description	The standard files (HTML pages) of the Coupler/Controller are extracted and written into the Flash.

Table 157: Register address 0x2043

Register address 0x2043 (8259$_{dez}$)	
Value	0x55AA
Access	Write
Description	Factory Settings

12.2.5.5 Firmware Information Registers

The following registers contain information on the firmware of the controller:

Table 158: Register address 0x2010

Register address 0x2010 (8208$_{dec}$) with a word count of 1	
Value	Revision, INFO_REVISION
Access	Read
Description	Firmware index, e.g. 0005 for version 5

Table 159: Register address 0x2011

Register address 0x2011 (8209$_{dec}$) with a word count of 1	
Value	Series code, INFO_SERIES
Access	Read
Description	WAGO serial number, e.g. 0750 for WAGO-I/O-SYSTEM 750

Table 160: Register address 0x2012

Register address 0x2012 (8210$_{dec}$) with a word count of 1	
Value	Order number, INFO_ITEM
Access	Read
Description	First part of WAGO order number, e.g. 841 for the controller 750-841 or 341 for the coupler 750-341 etc.

Table 161: Register address 0x2013

Register address 0x2013 (8211$_{dec}$) with a word count of 1	
Value	Major sub item code, INFO_MAJOR
Access	Read
Description	Firmware version Major Revision

Table 162: Register address 0x2014

Register address 0x2014 (8212$_{dec}$) with a word count of 1	
Value	Minor sub item code, INFO_MINOR
Access	Read
Description	Firmware version Minor Revision

Table 163: Register address 0x2020

Register address 0x2020 (8224$_{dec}$) with a word count of up to 16	
Value	Description, INFO_DESCRIPTION
Access	Read
Description	Information on the controller, 16 words

Table 164: Register address 0x2021

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

Table 165: Register address 0x2022

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

Table 166: Register address 0x2023

Register address 0x2023 (8227$_{dec}$) with a word count of up to 32	
Value	Description, INFO_LOADER_INFO
Access	Read
Description	Information to the programming of the firmware, 32 words

12.2.5.6 Constant Registers

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

Table 167: Register address 0x2000

Register address 0x2000 (8192$_{dec}$)	
Value	Zero, GP_ZERO
Access	Read
Description	Constant with zeros

Table 168: Register address 0x2001

Register address 0x2001 (8193$_{dec}$)	
Value	Ones, GP_ONES
Access	Read
Description	Constant with ones • –1 if this is declared as "signed int" • MAXVALUE if it is declared as "unsigned int"

Table 169: Register address 0x2002

Register address 0x2002 (8194$_{dec}$)	
Value	1,2,3,4, GP_1234
Access	Read
Description	This constant value is used to test the Intel/Motorola format specifier. If the master reads a value of 0x1234, then with Intel format is selected – this is the correct format. If 0x3412 appears, Motorola format is selected.

Table 170: Register address 0x2003

Register address 0x2003 (8195$_{dec}$)	
Value	Mask 1, GP_AAAA
Access	Read
Description	This constant is used to verify that all bits are accessible to the fieldbus master. This will be used together with register 0x2004.

Table 171: Register address 0x2004

Register address 0x2004 (8196$_{dec}$)	
Value	Mask 1, GP_5555
Access	Read
Description	This constant is used to verify that all bits are accessible to the fieldbus master. This will be used together with register 0x2003.

Table 172: Register address 0x2005

Register address 0x2005 (8197$_{dec}$)	
Value	Maximum positive number, GP_MAX_POS
Access	Read
Description	Constant in order to control arithmetic.

Table 173: Register address 0x2006

Register address 0x2006 (8198$_{dec}$)	
Value	Maximum negative number, GP_MAX_NEG
Access	Read
Description	Constant in order to control arithmetic

Table 174: Register address 0x2007

Register address 0x2007 (8199$_{dec}$)	
Value	Maximum half positive number, GP_HALF_POS
Access	Read
Description	Constant in order to control arithmetic

Table 175: Register address 0x2008

Register address 0x2008 (8200$_{dec}$)	
Value	Maximum half negative number, GP_HALF_NEG
Access	Read
Description	Constant in order to control arithmetic

Table 176: Register address 0x3000 to 0x5FFF

Register address 0x3000 to 0x5FFF (12288$_{dec}$ to 24575$_{dec}$)	
Value	Retain range
Access	Read/write
Description	These registers can be accessed as the flag/retain range

12.3 EtherNet/IP (Ethernet/Industrial Protocol)

12.3.1 General

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

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

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

CIP, as a major network independent standard, is already used with ControlNet and DeviceNet. Therefore, converting from one of these protocols to EtherNet/IP is easy to do. Data exchange takes place with the help of an object model.

In this way, ControlNet, DeviceNet and EtherNet/IP have the same application protocol and can therefore jointly use device profiles and object libraries. These objects enable plug-and-play interoperability between complex devices of different manufacturers.

12.3.2 Protocol overview in the OSI model

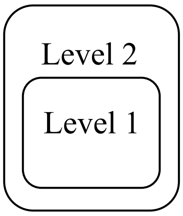
In order to clarify the interrelationships between DeviceNet, ControlNet and EtherNet/IP, the following diagram presents the associated ISO/OSI reference model.

Table 177: ISO/OSI reference model

<table>
<tr><td>7
Application Layer</td><td colspan="3">Object Library
(Communications, Applications, Time Synchronization)</td><td>Safety Object Library</td><td rowspan="3">Comon Industrial Protocol (CIP)</td></tr>
<tr><td>6
Presentation Layer</td><td colspan="3">Data Management Services
Explicit and I/O Messages</td><td>Safety Services and Messages</td></tr>
<tr><td>5
Session Layer</td><td colspan="4">Connection Management, Routing</td></tr>
<tr><td>4
Transport Layer</td><td>TCP/UDP</td><td rowspan="2">CompoNet
Network and Transport</td><td rowspan="2">ControlNet
Network and Transport</td><td rowspan="2">DeviceNet
Network and Transport</td><td rowspan="4">Network Adaptations of CIP</td></tr>
<tr><td>3
Network Layer</td><td>Internet Protocol</td></tr>
<tr><td>2
Data Link Layer</td><td>Ethernet
CSMA/CD</td><td>CompoNet
Time Slot</td><td>ControlNet
CTDMA</td><td>CAN
CSMA/NBA</td></tr>
<tr><td>1
Physical Layer</td><td>Ethernet</td><td>CompoNet</td><td>ControlNet</td><td>DeviceNet</td></tr>
</table>

12.3.3 Characteristics of the EtherNet/IP Protocol Software

The Ethernet/IP product classes are divided into 4 levels with each level containing a particular functionality. Each higher level in turn possesses at least the functionality of a lower level. The fieldbus coupler supports levels 1 and 2 of the Ethernet/IP product classes, which immediately build on each other.



Level 2: Level 1 + I/O Messages Server

Level 1: Explicit 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)

12.3.4 EDS File

The "Electronic Data Sheets" file (EDS file for short) contains the characteristics of the fieldbus coupler/controller and information regarding its communication capabilities. The EDS file required for Ethernet/IP operation is imported and installed by the corresponding configuration software.

Note

Downloading the EDS file!

You can download the EDS file in the download area of the WAGO web site:
http://www.wago.com → Service → Downloads → AUTOMATION

Information

Information about installing the EDS file

When installing the EDS file, refer to the information provided in the documentation of the configuration software, which you are using.

12.3.5 Object Model

12.3.5.1 General

For network communication, Ethernet/IP utilizes an object model in which all functions and data of a device are described.
Each node in the network is depicted as a collection of objects.
The object model contains terms that are defined as follows:

Object:
An object is an abstract representation of individual, related components within a device. It is determined by its data or attributes, its outwardly applied functions or services, and by its defined behavior.

Class:
A class describes a series of objects which all represent the same type of system components. A class is the generalization of an object. All objects in a class are identical as regards form and behavior, but can comprise differing attribute values.

Instance:
An instance describes a specific and physical occurrence of an object. The terms "object," "instance" and "object instance" all refer to a specific instance. Different instances of a class have the same services, the same behavior and the same variables (attributes). However, you can have different variable values.
For example, Finland is an instance of the "Land" object class.

Variable:
The variables (attributes) describe an externally visible characteristic or the function of an object. Typical attributes include configuration or status information.
For example, the ASCII name of an object or the repetition frequency of a periodic object is output.

Service:
A service is a function supported by an object and/or an object class. CIP defines a group of common services that are applied to the attributes. These services execute specified actions.
Example: Reading variables.

Behavior:
The behavior specifies how an object functions. The functions result from various occurrences, which are determined by the object, e.g. receiving service requests, recording internal errors or the sequence of timers.

12.3.5.2 Class Overview

CIP classes are included in the CIP specification of ODVA. They describe the properties (Volume 1, "Common Industrial Protocol") of Ethernet and CAN independent of their physical interface. The physical interface is described in a separate specification. For Ethernet/IP, this is Volume 2 ("Ethernet/IP Adaptation of CIP"), which describes the adaption of Ethernet /IP to CIP.

For this purpose, WAGO uses classes 01_{hex}, 02_{hex}, 04_{hex}, 05_{hex}, 06_{hex} and $F4_{hex}$, which are described in Volume 1 ("Common Industrial Protocol").
Classes $F5_{hex}$ and $F6_{hex}$ are supported from Volume 2 ("Ethernet/IP Adaptation of CIP").

WAGO-specific classes listed in the overview table below are also available.

All CIP Common classes listed and the WAGO-specific classes listed below that are described in detail in the following individual sections after a brief explanation of the table headings in the object descriptions.

Table 178: CIP common class

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

Table 179: WAGO specific classes

Class	Name
64 hex	Coupler/Controller Configuration Object
65 hex	Discrete Input Point
66 hex	Discrete Output Point
67 hex	Analog Input Point
68 hex	Analog Output Point
69 hex	Discrete Input Point Extended 1
6A hex	Discrete Output Point Extended 1
6B hex	Analog Input Extended 1
6C hex	Analog Output Point Extended 1
6D hex	Discrete Input Point Extended 2
6E hex	Discrete Output Point Extended 2
6F hex	Analog Input Point Extended 2
70 hex	Analog Output Point Extended 2
71 hex	Discrete Input Point Extended 3
72 hex	Discrete Output Point Extended 3
73 hex	Analog Input Point Extended 3
74 hex	Analog Output Point Extended 3
80 hex	Module Configuration
81 hex	Module Configuration Extended 1
A0 hex	Input fieldbus variable USINT
A1 hex	Input fieldbus variable USINT Extended 1
A2 hex	Input fieldbus variable USINT Extended 2
A3 hex	Output fieldbus variable USINT
A4 hex	Output fieldbus variable USINT Extended 1
A5 hex	Output fieldbus variable USINT Extended 2
A6 hex	Input fieldbus variable UINT
A7 hex	Input fieldbus variable UINT Extended 1
A8 hex	Output fieldbus variable UINT
A9 hex	Output fieldbus variable UINT Extended 1
AA hex	Input fieldbus variable UDINT
AB hex	Input fieldbus variable UDINT Offset UINT
AC hex	Output fieldbus variable UDINT
AD hex	Output fieldbus variable UDINT Offset UINT

12.3.5.3 Explanation of the Table Headings in the Object Descriptions

Table 180: Explanation of the table headings in the object descriptions

Table heading	Description
Attribute ID	Integer value which is assigned to the corresponded attribute
Access	**Set:** The attribute can be accessed by means of Set_Attribute services. **Note** **Response also possible with Get_Attribute service!** 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. **Get_Attribute_All:** Delivers content of all attributes. **Set_Attribute_Single:** Modifies an attribute value. **Reset:** Performs a restart. 0: Restart 1: Restart and restoration of factory settings
NV	**NV (non volatile):** The attribute is permanently stored in the controller.
	V (volatile): The attribute is not permanently stored in the controller. **Note** **Without specifying, the attribute is not saved!** If this column is missing, all attributes have the type V (volatile).
Name	Designation of the attribute
Data type	Designation of the CIP data type of the attribute
Description	Short description for the Attribute
Default value	Factory settings

12.3.5.4 Identity (01 $_{hex}$)

The "Identity" class provides general information about the fieldbus coupler/controller that clearly identifies it.

Instance 0 (Class Attributes)

Table 181: Identity (01 $_{hex}$) – Class

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

Instance 1

Table 182: Identity (01 hex) – Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Vendor ID	UINT	Manufacturer identification	40 (0x0028)
2	Get	Device Type	UINT	General type designation of the product	12 (0x000C)
3	Get	Product Code	UINT	Designation of the coupler/ controller	z. B. 841 (0x0349), 873 (0x0369), 341(0x0155) etc.
4	Get	Revision	STRUCT of:	Revision of the identity objects	Depending on the firmware
		Major Revision	UINT		
		Minor Revision	UINT		
5	Get	Status	WORD	Current status of the device	Bit 0 Assignment to a master Bit 1 = 0 reserved Bit 2 (configured) = 0 Configuration is unchanged = 1 Configuration is different to the manufacturers parameters Bit 3 = 0 reserved Bit 4-7 Extended Device Status =0010 at least one faulted I/O connection =0011 no I/O connection established Bit 8-11 not used Bit 12-15 =0 reserved
6	Get	Serial Number	UINT	Serial number	The last 4 digits of MAC ID
7	Get	Product Name	SHORT_ STRING	Product name	

Common Services

Table 183: Identity (01 hex) – Common service

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

12.3.5.5 Message Router (02 hex)

The "Message Router Object" provides connection points (in the form of classes or instances), which can use a client for addressing services (reading, writing). These messages can be transmitted both when connected and when unconnected from the client to the fieldbus coupler.

Instance 0 (Class Attributes)

Table 184: Message router (02 hex) – Class

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

Note

Get_Attribute_All service can only be used!
The class attributes are only accessible with the Get_Attribute_All service.

Instance 1

Table 185: Message router (02 hex) – Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	ObjectList	STRUCT of:	-	
		Number	UINT	Number of implemented classes	40 (0x0028)
		Classes	UINT	Implemented classes	01 00 02 00 04 00 06 00 F4 00 F5 00 F6 00 64 00 65 0066 0067 00 68 00 69 00 6A 00 6B 00 6C 00 6D 00 6E 00 6F 00 70 00 71 00 72 00 73 00 74 00 80 00 81 00 A0 00 A1 00 A2 00 A6 00 A7 00 AA 00 AB 00 A3 00 A4 00 A5 00 A8 00 A9 00 AC 00 AD 00
2	Get	NumberAvailable	UINT	Maximum number of different connections	128 (0x0080)

Common Services

Table 186: Message router (02 hex) – Common service

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

12.3.5.6 Assembly Object (04 hex)

Using the "Assembly" classe, even several diverse objects can be combined. These could be, for example, input and output data, status and control information or diagnostic information. WAGO uses the manufacturer-specific instances in order to provide these objects for you in various arrangements. This gives you an efficient way to exchange process data. The following is a description of the individual static Assembly instances with their contents and arrangements.

Instance (Class Attributes)

Table 187: Assembly (04 hex) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	2 (0x0002)
2	Get	Max Instance	UINT	Highest Instance	111 (0x006F)

Overview of static Assembly instances

Table 188: Overview of static Assembly instances

Instance	Description
Instance 101 (65 hex)	For analog and digital output data, as well as fieldbus input variables
Instance102 (66 hex)	For digital output data and fieldbus input variables
Instance 103 (67 hex)	For analog output data and fieldbus input variables
Instance 104 (68 hex)	For analog and digital intput data, status and fieldbus output variables
Instance 105 (69 hex)	For digital input data, status and fieldbus output variables
Instance 106 (6A hex)	For analog input data, status and fieldbus output variables
Instance 107 (6B hex)	For digital and analog input data and fieldbus output variables
Instance 108 (6C hex)	For digital input data and fieldbus output variables
Instance109 (6D hex)	For analog input data and fieldbus output variables
Instance 110 (6E hex)	For fieldbus output variables
Instance 111 (6F hex)	For fieldbus intput variables

Instance 101 (65 hex)

This assembly instance contains analog and digital output data. Any fieldbus input variables that may be defined are attached behind this.

Table 189: Static assembly instances – Instance 101 (65 $_{hex}$)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get/Set	Data	ARRAY of BYTE	Only analog and digital output data, as well as possible fieldbus input variables, are contained in the process image.	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

Instance 102 (66 $_{hex}$)

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

Table 190: Static assembly instances – Instance 102 (66 $_{hex}$)

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

Instance 103 (67 $_{hex}$)

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

Table 191: Static assembly instances – Instance 103 (67 $_{hex}$)

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

Instance 104 (68 $_{hex}$)

This assembly instance contains analog and digital input data, status (= value from class 100, instance 1, attribute 5) and fieldbus output variables.

Table 192: Static assembly instances – Instance 104 (68 $_{hex}$)

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

Instance 105 (69 $_{hex}$)

This assembly instance contains only digital input data, status (= value from class 100, instance 1, attribute 5) and fieldbus output variables.

Table 193: Static assembly instances – Instance 105 (69 $_{hex}$)

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

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

This assembly instance contains only analog input data, status (= value from class 100, instance 1, attribute 5) and fieldbus output variables.

Table 194: Static assembly instances – Instance 106 (6A $_{hex}$)

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

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

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

Table 195: Static assembly instances – Instance 107 (6B $_{hex}$)

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

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

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

Table 196: Static assembly instances – Instance 108 (6C $_{hex}$)

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

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

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

Table 197: Static assembly instances – Instance 109 (6C $_{hex}$)

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

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

This assembly instance contains fieldbus output variables.

Table 198: Static assembly instances – Instance 110 (6E $_{hex}$)

Attribute ID	Access	Name	Data type	Description	Default value
3	Get	Data	ARRAY of BYTE	Reference of the process image: only PFC output variables	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

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

This assembly instance contains fieldbus input variables only.

Table 199: Static assembly instances – Instance 111 (6F $_{hex}$)

Attribute ID	Access	Name	Data type	Description	Default value
3	Set	Data	ARRAY of BYTE	Reference of the process image: only PFC input variables	-
4	Get	Data Size	UNIT	Number of Bytes in the process data image	-

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

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

Instance 199 (C7 $_{hex}$) “Listen only”

This instance is used to establish a connection based on an existing exclusive owner connection. The new connection also has the same transmission parameters as the exclusive owner connection. When the exclusive owner connection is cleared, this connection, too, is automatically cleared. The data length of this instance is always zero.

This instance can only be used in the “consumed path” (from the point of view of the slave device).

Common Service

Table 200: Static assembly instances – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value

The software inspects the writing of attribute 3 of assembly instances 101, 102 and 103. If the limit value has been exceeded, it is identified and, if necessary, corrected. However, a write request is not rejected. This means that if less data is received than expected, only this data is written. If more data is received than expected, the received data at the upper limit is deleted. In the case of explicit messages, however, a defined CIP is generated even though the data has been written.

12.3.5.7 Connection (05 $_{hex}$)

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

12.3.5.8 Connection Manager (06 $_{hex}$)

The "Connection Manager Object" provides the internal resources that are required for the input and output data and explicit messages. In addition, the administration of this resource is an assignment of the "Connection Manager Object".

For each connection (input and output data or explicit), another instance of the connection class is created. The connection parameters are extracted from the "Forward Open" service, which is responsible for establishing a connection.

The following services are supported for the first instance:

- Forward_Open
- Unconnected_Send
- Forward_Close

No class and instance attributes are visible.

12.3.5.9 Port Class (F4 hex)

The "Port Class Object" specifies the existing CIP ports on the fieldbus coupler/coupler. There is one instance for each CIP port.

Instance 0 (Class Attributes)

Table 201: Port class (F4 hex) – Class

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

Instance 1

Table 202: Port class (F4 hex) – Instance 1

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Port Type	UINT	-	4 (0x0004)
2	Get	V	Port Number	UINT	CIP port number	2 (0x0002) (EtherNet/IP)
3	Get	V	Port Object	UINT	Number of 16 bit words in the following path	2 (0x0002)
				Padded EPATH	Object, which manages this port	0x20 0xF5 0x24 0x01 (equals TCP/IP Interface Object)
4	Get	V	Port Name	SHORT_ STRING	Port name	“”
7	Get	V	Node Address	Padded EPATH	Port segment (IP address)	Depends on IP address

Common Services

Table 203: Port class (F4 $_{hex}$) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
01 $_{hex}$	Yes	Yes	Get_Attribute_All	Supplies contents of all attributes
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.10 TCP/IP Interface ($F5_{hex}$)

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

The underlying physical communications interface that is connected with the TCP/IP interface object can be any interface supported by the TCP/IP protocol. Examples of components that can be connected to a TCP/IP interface object include the following: an Ethernet interface 802.3, an ATM (Asynchronous Transfer Mode) interface or a serial interface for protocols such as PPP (Point-to-Point Protocol).
The TCP/IP interface object provides an attribute, which is identified by the link-specific object for the connected physical communications interface. The link-specific object should typically provide link-specific counters as well as any link-specific configuration attributes.

Each device must support exactly one instance of the TCP/IP interface object for each TCP/IP-compatible communications interface. A request for access to the first instance of the TCP/IP interface object must always refer to the instance connected with the interface, which is used to submit the request.

Instance 0 (Class Attributes)

Table 204: TCP/IP interface ($F5_{hex}$) – Class

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

Instance 1

Table 205: TCP/IP interface ($F5_{hex}$) – Instance 1

Attribute ID	Access	NV	Name	Data type	Description	Default value
1	Get	V	Status	DWORD	Interface state	-
2	Get	V	Configuration Capability	DWORD	Interface flags for possible kinds of configuration	0x00000017
3	Set	NV	Configuration Control	DWORD	Specifies, how the device gets is TCP/IP configuration after the first Power On	0x00000011
4	Get	V	Physical Link Object	STRUCT of		
			Path size	UINT	Number of 16 Bit words in the following path	0x0002
			Path	Padded EPATH	Logical path, which points to the physical Link object	0x20 0xF6 0x24 0x03 (equates to the Ethernet Link Object)
5	Set	NV	Interface Configuration	STRUCT of	-	
			IP Address	UDINT	IP address	0
			Network Mask	UDINT	Net work mask	0
			Gateway Address	UDINT	IP address of default gateway	0
			Name Server	UDINT	IP address of the primary name of the server	0
			Name Server 2	UDINT	IP address of the secondary name of the server	0
			Domain Name	STRING	Default domain name	“”
6	Set	NV	Host Name	STRING	Device name	“”

Common Services

Table 206: TCP/IP interface ($F5_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
01_{hex}	Yes	Yes	Get_Attribute_All	Supplies contents of all attributes
$0E_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10_{hex}	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.11 Ethernet Link ($F6_{hex}$)

The "Ethernet Link Object" contains link-specific counter and status information for an Ethernet 802.3 communications interface. Each device must support exactly one instance of the Ethernet Link Object for each Ethernet IEEE 802.3 communications interface on the module. An Ethernet link object instance for an internal interface can also be used for the devices, e.g. an internal port with an integrated switch.

Instance 0 (Class Attributes)

Table 207: Ethernet link ($F5_{hex}$) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	3 (0x0003)
2	Get	Max Instance	UDINT	Max. number of instances	3 (0x0003)
3	Get	Num Instances	UDINT	Number of the current instanced connections	3 (0x0003)

Instance 1

Table 208: Ethernet link (F6 $_{hex}$) – Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Interface Speed	UDINT	Transfer rate	10 (0x0A) or 100 (0x64)
2	Get	Interface Flags	DWORD	Interface configuration and status information Bit 0: Link status Bit 1: Half/full lduplex Bit 2…4: Detection status Bit 5: Manual settings require reset Bit 6: Local hardware error Bit 7…31: Reserved	Value is dependent upon Ethernet connection.
3	Get	Physical Address	ARRAY of 6 UINTs	MAC layer address	MAC ID of the device
6	Set	Interface Control	STRUCT of:	Configuration of the physical interface	-
		Control Bits	WORD	Interface configuration bits Bit 0: Automatic detection Bit 1: Default duplex mode Bit 2…15: Reserved	0x0001
		Forced Interface Speed	UINT	Preset interface speed	10 (0x000A) or 100 (0x0064)
7	Get	Interface Type	USINT	Interface type Value 0: Unknown Value 1: Internal interface; e.g., in the case of an integrated switch Value 2: Twisted pair (e.g. 100Base-TX). Value 3: fiber glass (e.g. 100Base-FX). Value 4…256: Reserved	2 (0x02) – Twisted Pair
8	Get	Interface Status	USINT	Interface status Value 0: Unknown Value 1: Interface active and ready to send/receive. Value 2: Interface deactivated. Value 3: Interface is testing Wert 4…256: Reserved	-

Table 208: Ethernet link (F6 $_{hex}$) – Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
9	Get/ Set	Admin Status	USINT	Admin status: Value 0: Reserved Value 1: Interface active Value 2: Interface deactivated. Is this the only CIP interface, a request for deactivation will be receipted with error code 0x09 Value 3…256: Reserved	1 (0x01)
10	Get	Interface Label	SHORT_ STRING	Name of the interface	“Port 1“

Instance 2 – Port 2

Table 209: Ethernet link (F6 $_{hex}$) – Instance 2

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Interface Speed	UDINT	Transfer rate	10 (0x0000000A) or 100 (0x00000064)
2	Get	Interface Flags	DWORD	Interface configuration and status information Bit 0: Link status Bit 1: Half/full lduplex Bit 2…4: Detection status Bit 5: Manual settings require reset Bit 6: Local hardware error Bit 7…31: Reserved	Value is dependent upon Ethernet connection.
3	Get	Physical Address	ARRAY of 6 UINTs	MAC layer address	MAC-ID des Fieldbus couplers/ controllers
6	Set	Interface Control	STRUCT of:	Configuration of the physical interface	-
		Control Bits	WORD	Interface configuration bits Bit 0: Automatic detection Bit 1: Default duplex mode Bit 2…15: Reserved	0x0001
		Forced Interface Speed	UINT	Preset interface speed	10 (0x000A) or 100 (0x0064)
7	Get	Interface Type	USINT	Interface type Value 0: Unknown Value 1: Internal interface; e.g., in the case of an integrated switch Value 2: Twisted pair (e.g. 100Base-TX). Value 3: fiber glass (e.g. 100Base-FX). Value 4…256: Reserved	2 (0x02) – Twisted Pair
8	Get	Interface Status	USINT	Interface status Value 0: Unknown Value 1: Interface active and ready to send/receive. Value 2: Interface deactivated. Value 3: Interface is testing Wert 4…256: Reserved	-

Table 209: Ethernet link (F6 $_{hex}$) – Instance 2

Attribute ID	Access	Name	Data type	Description	Default value
9	Get/ Set	Admin Status	USINT	Admin status: Value 0: Reserved Value 1: Interface active Value 2: Interface deactivated. Is this the only CIP interface, a request for deactivation will be receipted with error code 0x09 Value 3…256: Reserved	1 (0x01)
10	Get	Interface Label	SHORT_ STRING	Name of the interface	“Port 2“

Instance 3 – Internal Port 3

Table 210: Ethernet link (F6 $_{hex}$) – Instance 3

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Interface Speed	UDINT	Transfer rate	100 (0x64)
2	Get	Interface Flags	DWORD	Interface configuration and status information	3 (0x03) – Link active (Bit 0), Full duplex (Bit 1)
3	Get	Physical Address	ARRAY of 6 UINTs	MAC layer address	MAC ID of the device
6	Set	Interface Control	STRUCT of:	Configuration of the physical interface	-
		Control Bits	WORD	Interface configuration bits	3 (0x03) – Link active (Bit 0), Full duplex (Bit 1)
		Forced Interface Speed	UINT	Baud rate	100 (0x64)
7	Get	Interface Type	UINT	Interface type	1 (0x01) – internal Port
8	Get	Interface Status	UINT	Interface status	1 (0x01) – active
9	Get	Admin Status	UINT	Admin status	1 (0x01) – active
10	Get	Interface Label	SHORT_ STRING	Name of the interface	„Internal Port 3“

Common Services

Table 211: Ethernet link (F6 $_{hex}$) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
01 $_{hex}$	Yes	Yes	Get_Attribute_All	Supplies contents of all attributes
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value

Note

Changes with service ”Set_Attribute_Single“ not directly effective!
Attributes (particularly the attributes 6 and 9) which were changed over the service “Set_Attribute_Single“, become only effective after the next Power-On-Reset of the controller.

12.3.5.12 Coupler/Controller Configuration (64 $_{hex}$)

The fieldbus coupler configuration class allows reading and configuration of some important fieldbus/controller process parameters. The following listings explain in details all supported instances and attributes.

Instance 0 (Class Attributes)

Table 212: Coupler/Controller configuration (64 $_{hex}$) – Class

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

Instance 1

Table 213: Coupler/Controller configuration (64 $_{hex}$) – Instance 1

Attribute ID	Access	NV	Name	Data type	Description	Default value
5 (0x05)	Get	V	ProcessState	USINT	State of coupler/controller, error mask: Bit 0: Internal bus error Bit 3: Module diagnostics (0x08) Bit 7: Fieldbus error (0x80)	0
6 (0x06)	Get	V	DNS_i_ Trmnldia	UINT	Module diagnostics: Bit 0..7: Module number Bit 8..14: Module channel Bit 15: 0/1 Error, repair/arisen	0
7 (0x07)	Get	V	CnfLen. AnalogOut	UINT	Number of I/O bits for the analog output	-
8 (0x08)	Get	V	CnfLen. AnalogInp	UINT	Number of I/O bits for the analog input	-
9 (0x09)	Get	V	CnfLen. DigitalOut	UINT	Number of I/O bits for the digital output	-
10 (0x0A)	Get	V	CnfLen. DigitalInp	UINT	Number of I/O bits for the digital input	-
11 (0x0B)	Set	NV	Bk_Fault_ Reaction	USINT	Fieldbus error reaction 0: stop local I/O cycles 1: set all output to 0 2: no error reaction 3: no error reaction 4: PFC task takes over control of the outputs (apply to controllers)	1
12..26 (0x0C...0x1A)	Reserved for compatibility to DeviceNet					
40..43 (0x28...0x2B)	Reserved for compatibility to DeviceNet					
45 (0x2D)	Get	V	Bk_Led_Err _Code	UINT	I/O LED error code	0
46 (0x2E)	Get	V	Bk_Led_Err _Arg	UINT	I/O LED error argument	0

Attribute ID	Access	NV	Name	Data type	Description	Default value
47 (0x2F)	Get	V	Bk_Diag_Value	UINT	Contains the diagnostic byte Note: This attribute has to be read out before attribute 6 (DNS_i_Trmnldia), because during the reading of attribute 6 the diagnostic byte contains the data of the next diagnostic	0
100 (0x64)	Set	NV	Bk_FbInp_Var _Cnt	UINT	Determines the number of bytes for the PFC input fieldbus variables, which are added to the assembly object. This number is added to the consuming path. assembly instances (101...103)	0
101 (0x65)	Set	NV	Bk_FbOut_Var _Cnt	UINT	Determines the number of bytes for the PFC output fieldbus variables, which are added to the assembly object. This number is added to the producing path. assembly instances (104...109)	0
102 (0x66)	Set	NV	Bk_FbInp_Plc Only_Var_Cnt	UINT	Determines the number of bytes for the PFC input fieldbus variables, which are received via assembly instance 111.	4
103 (0x67)	Set	NV	Bk_FbInp_Start Plc_Var_Cnt	UINT	Determines starting from which position the PFC input fieldbus variables for the assembly instance 111 to be received.	0
104 (0x68)	Set	NV	Bk_FbOut_Plc Only_Var_Cnt	UINT	Determines the number of bytes for the PFC output fieldbus variables, which are received via assembly instance 110.	4
105 (0x69)	Set	NV	Bk_FbOut_Star t Plc_Var_Cnt	UINT	Determines starting from which position the PFC output fieldbus variables for the assembly instance 110 to be received.	0

120 (0x78)	Set	NV	Bk_Header CfgOT	UINT	Indicates whether the RUN/IDLE header is used originator → target direction 0: is used 1: is not used	0x0000
121(0x79)	Set	NV	Bk_Header CfgTO	UINT	Indicates whether the RUN/IDLE header is used originator → target direction 0: is used 1: is not used	0x0001

Common Service

Table 214: Coupler/Controller configuration (64 $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value

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

This class allows the reading of data of a particular digital input point.

Instance 0 (Class-Attributes)

Table 215: Discrete input point (65 $_{hex}$) – Class

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

Instance 1 ... 255 (Digital output value 1 up to 255)

Table 216: Discrete input point (65 $_{hex}$) – Instance 1...255

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

Common Services

Table 217: Discrete input point (65 $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.14 Discrete Input Point Extended 1 (69 $_{hex}$)

The extension of the "Discrete Input Point" class enables the reading of data from a fieldbus node that contains over 255 digital input points (DIPs). The instance scope of the "Discrete Input Point Extended 1" class covers DIPs from 256 to 510 in the fieldbus node.

Instance 0 (Class Attributes)

Table 218: Discrete Input Point Extended 1(69 hex,) – Class

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

Instance 256 ... 510 (Digital input value 256 up to 510)

Table 219: Discrete output point (66 $_{hex}$) – Instance 256...510

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

Common Services

Table 220: Discrete Input Point Extended 1 (69 $_{hex}$) – Common service

Service code	Service available		Service-name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.15 Discrete Input Point Extended 2 (6D $_{hex}$)

The extension of the "Discrete Input Point" class enables the reading of data from a fieldbus node that contains over 510 digital input points (DIPs). The instance scope of the "Discrete Input Point Extended 2" class covers DIPs from 511 to 765 in the fieldbus node.

Instance 0 (Class Attributes)

Table 221: Discrete Input Point Extended 2 (6D hex) – Class

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

Instance 511 ... 765 (Digital input value 511 up to 765)

Table 222: Analog input point (67 $_{hex}$) – Instance 1

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	ARRAY of BYTE	Analog input	-
2	Get	AipObj_Value_Length	USINT	Length of the input data AipObj_Value (in byte)	-

Common Services

Table 223: Analog input point (67 $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.16 Discrete Input Point Extended 3 (71 $_{hex}$)

The extension of the "Discrete Input Point" class enables the reading of data from a fieldbus node that contains over 765 digital input points (DIPs). The instance scope of the "Discrete Input Point Extended 3" class covers DIPs from 766 to 1020 in the fieldbus node.

Instance 0 (Class-Attributes)

Table 224: Discrete Input Point Extended 3 (71 hex) – Class

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

Instance 766 ... 1020 (Digital input value 766 up to 1020)

Table 225: Discrete Input Point Extended 3 (71 $_{hex}$) – Instance 766...1020

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

Common Services

Table 226: Discrete Input Point Extended 3 (71 $_{hex}$) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.17 Discrete Output Point (66 $_{hex}$)

This class enables data exchange for a particular digital output point.

Instance 0 (Class Attributes)

Table 227: Discrete Output Point (66 hex) – Class

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

Instance 1 ... 255 (Digital output value 1 up to 255)

Table 228: Discrete Output Point (66 hex) – Instance 1...255

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

Common Services

Table 229: Discrete Output Point (66 hex) – Common service

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

12.3.5.18 Discrete Output Point Extended 1 (6A hex)

The extension of the "Discrete Output Point" class enables the exchange of data from a fieldbus node that contains over 255 digital output points (DOPs). The instance scope of the "Discrete Output Point Extended 1" class covers DOPs from 256 to 510 in the fieldbus node.

Instance 0 (Class Attributes)

Table 230: Discrete Output Point Extended 1 (6A hex) – Class

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

Instance 256 ... 510 (Digital output value 256 up to 510)

Table 231: Discrete Output Point Extended 1 (6A hex) – Instance 256...510

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

Common Services

Table 232: Discrete Output Point Extended 1 (6A hex) – Common service

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

12.3.5.19 Discrete Output Point Extended 2 (6E $_{hex}$)

The extension of the "Discrete Output Point" class enables the exchange of data from a fieldbus node that contains over 510 digital output points (DOPs). This instance cope of the "Discrete Output Point Extended 1" class covers the DOPs from 511 to 765 in the fieldbus node.

Instance 0 (Class Attributes)

Table 233: Discrete Output Point Extended 2 (6E $_{hex}$) – Class

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

Instance 511 ... 765 (Digital output value 511 up to 765)

Table 234: Discrete Output Point Extended 2 (6E $_{hex}$) – Instance 511...765

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

Common Services

Table 235: Discrete Output Point Extended 2 (6E $_{hex}$) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.20 Discrete Output Point Extended 3 (72 $_{hex}$)

The extension of the "Discrete Output Point" class enables the exchange of data from a fieldbus node that contains over 765 digital output points (DOPs). The instance scope of the "Discrete Output Point Extended 2" class covers DOPs from 766 to 1020 in the fieldbus node.

Instance 0 (Class Attributes)

Table 236: Discrete Output Point Extended 3 (72 $_{hex}$) – Class

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

Instance 766 ... 1020 (Digital Output value 766 up to 1020)

Table 237: Discrete Output Point Extended 3 (72 $_{hex}$) – Instance 766...1020

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

Common Services

Table 238: Discrete Output Point Extended 2 (6E $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.21 Analog Input Point (67 $_{hex}$)

This class enables the reading of data of a particular analog input point (AIP). An analog input point is part of an analog input module.

Instance 0 (Class Attributes)

Table 239: Analog Input Point (67 $_{hex}$) – Class

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

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

Table 240: Analog Input Point (67 $_{hex}$) – Instance 1 ... 255

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	ARRAY of BYTE	Analog Input	-
2	Get	AipObj_Value_ Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 241: Analog Input Point (67 $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.22 Analog Input Point Extended 1 (6B hex)

The extension of the "Analog Input Point" class enables the reading of data from a fieldbus node that contains over 255 analog outputs (AIPs). The instance scope of the "Analog Input Point Extended 1" class covers AIPs from 256 to 510 in the fieldbus node.

Instance 0 (Class Attributes)

Table 242: Analog Input Point Extended 1 (6B hex) – Class

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

Instance 256 ... 510 (Analog Input value 256 up to 510)

Table 243: Analog Input Point Extended 1 (6B hex) – Instance 256 ... 510

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	ARRAY of BYTE	Analog Input	-
2	Get	AipObj_Value_ Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 244: Analog Input Point Extended 1 (6B hex) – Common service

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

12.3.5.23 Analog Input Point Extended 2 (6F hex)

The extension of the "Analog Input Point" class enables the reading of data from a fieldbus node that contains over 510 analog outputs (AIPs). The instance scope of the "Analog Input Point Extended 2" class covers AIPs from 511 to 765 in the fieldbus node.

Instance 0 (Class Attributes)

Table 245: Analog Input Point Extended 2 (6F hex) – Class

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

Instance 511 ... 765 (Analog Input 511 up to 765)

Table 246: Analog Input Point Extended 2 (6F $_{hex}$) – Instance 511 ... 765

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	ARRAY of BYTE	Analog Input	-
2	Get	AipObj_Value_ Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 247: Analog Input Point Extended 2 (6F $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.24 Analog Input Point Extended 3 (73 $_{hex}$)

The extension of the "Analog Input Point" class enables the reading of data from a fieldbus node that contains over 765 analog outputs (AIPs). The instance scope of the "Analog Input Point Extended 3" class covers AIPs from 766 to 1020 in the fieldbus node.

Instance 0 (Class Attributes)

Table 248: Analog Input Point Extended 3 (73 $_{hex}$) – Class

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

Instance 766 ... 1020 (Analog input value 766 up to 1020)

Table 249: Analog Input Point Extended 3 (73 $_{hex}$) – Instance 766 ... 1020

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AipObj_Value	ARRAY of BYTE	Analog Input	-
2	Get	AipObj_Value_ Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 250: Analog Input Point Extended 3 (73 hex) – Common service

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

12.3.5.25 Analog Output Point (68 hex)

This class enables the reading of data of a particular analog output point (AOP). An analog output point is part of an analog output module.

Instance 0 (Class Attributes)

Table 251: Analog Output Point (68 hex) – Class

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

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

Table 252: Analog Output Point (68 hex) – Instance 1...255

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AopObj_Value	ARRAY of BYTE	Analog Output	-
2	Get	AopObj_Value _Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 253: Analog Output Point (68 hex) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.26 Analog Output Point Extended 1 (6C hex)

The extension of the "Analog Output Point" class enables the exchange of data from a fieldbus node that contains over 255 analog output points (AOPs). The instance scope of the "Discrete Output Point Extended 1" class covers AOPs from 256 to 510 in the fieldbus node.

Instance 0 (Class Attributes)

Table 254: Analog Output Point Extended 1 (6C hex) – Class

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

Instance 256 ... 510 (Analog output value 256 up to 510)

Table 255: Analog Output Point Extended 1 (6C hex) – Instance 256...510

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AopObj_Value	ARRAY of BYTE	Analog Output	-
2	Get	AopObj_Value _Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 256: Analog Output Point Extended 1 (6C hex) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.27 Analog Output Point Extended 2 (70 hex)

The extension of the "Analog Output Point" class enables the exchange of data from a fieldbus node that contains over 510 analog output points (AOPs). The instance scope of the "Discrete Output Point Extended 2" class covers AOPs from 511 to 765 in the fieldbus node.

Instance 0 (Class Attributes)

Table 257: Analog Output Point Extended 2 (70 hex) – Class

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

Instance 511 ... 765 (Analog output value 511 up to 765)

Table 258: Analog Output Point Extended 2 (70 hex) – Instance 511...765

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AopObj_Value	ARRAY of BYTE	Analog Output	-
2	Get	AopObj_Value _Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 259: Analog Output Point Extended 2 (70 hex) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.28 Analog Output Point Extended 3 (74 hex)

The extension of the "Analog Output Point" class enables the exchange of data from a fieldbus node that contains over 765 analog output points (AOPs). The instance scope of the "Discrete Output Point Extended 3" class covers AOPs from 766 to 1020 in the fieldbus node.

Instance 0 (Class Attributes)

Table 260: Analog Output Point Extended 3 (74 hex) – Class

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

Instance 766 ... 1020 (Analog output value 766 up to 1020)

Table 261: Analog Output Point Extended 3 (74 hex) – Instance 766...1020

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	AopObj_Value	ARRAY of BYTE	Analog Output	-
2	Get	AopObj_Value _Length	USINT	Length of the output data AopObj_Value (in byte)	-

Common Services

Table 262: Analog Output Point Extended 3 (74 hex) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E hex	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 hex	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.29 Module Configuration (80 hex)

Instance 0 (Class Attributes)

Table 263: Module Configuration (80 hex) – Class

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

Instance 1 … 255 (Clamp 0 up to 254)

Table 264: Module Configuration (80 hex) – Instance 1...255

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

Common Services

Table 265: Module Configuration (80 hex) – Common service

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

12.3.5.30 Module Configuration Extended (81 hex)

The same as "Module Configuration (80 hex)" but with a description of module 255.

Instance 0 (Class Attributes)

Table 266: Module Configuration Extended (81 $_{hex}$) – Class

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

Instance 256 (Clamp 255)

Table 267: Module Configuration Extended (81 $_{hex}$) – Instance 256

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

Common Services

Table 268: Module Configuration Extended (81 $_{hex}$) – Common service

Service code	Service available		Service-Name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.31 Input Fieldbus Variable USINT (A0 $_{hex}$)

The class enables the reading of data from a particular PLC input variable.

For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for output variables %QB2552...%QB2806.

Instance 0 (Class Attributes)

Table 269: Input fieldbus variable USINT (A0 $_{hex}$) – Class

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

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

Table 270: Input fieldbus variable USINT (A0 $_{hex}$) – Instance 1...255

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

Common Services

Table 271: Input fieldbus variable USINT (A0 $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.32 Input Fieldbus Variable USINT Extended 1 (A1 $_{hex}$)

The extension of the "Input Fieldbus Variable USINT" class enables the reading of PLC input variable data. The instance scope of the "Input Fieldbus Variable USINT Extended 1" class covers the PLC input variable data from 256 to 510.

For WAGO-I/O-*PRO* or CoDeSys, that means the PLC addresses for input variables %IB2807...%IB3061.

Instance 0 (Class Attributes)

Table 272: Input Fieldbus Variable USINT Extended 1 (A1 $_{hex}$) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x00FF)

Instance 256...510 (Input variable 256 up to 510)

Table 273: Input fieldbus variable USINT Extended 1 (A1 $_{hex}$) – Instance 256...510

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

Common Services

Table 274: Input fieldbus variable USINT Extended 1 (A1 $_{hex}$) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value.

12.3.5.33 Input Fieldbus Variable USINT Extended 2 (A2 $_{hex}$)

The extension of the "Input Fieldbus Variable USINT" class enables the reading of PLC input variable data. The instance scope of the "Input Fieldbus Variable USINT Extended 1" class covers the PLC input variable data from 256 to 510. For WAGO-I/O-*PRO* or CoDeSys, that means the PLC addresses for input variables %IB2807...%IB3061.

Instance 0 (Class Attributes)

Table 275: Input Fieldbus Variable USINT Extended 2 (A2 $_{hex}$) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	2 (0x0002)

Instance 511...512 (Input variable 511 up to 512)

Table 276: Input Fieldbus Variable USINT Extended 2 (A2 $_{hex}$) – Instance 511...512

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

Common Services

Table 277: Input fieldbus variable USINT Extended 2 (A2 $_{hex}$) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.34 Output Fieldbus Variable USINT (A3 $_{hex}$)

The class enables the exchange of data from a particular PLC output variable.

For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for output variables %QB2552...%QB2806.

Instance 0 (Class Attributes)

Table 278: Output fieldbus variable USINT (A3 $_{hex}$) – Class

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

Instance 1...255 (Output variables 1 up to 255)

Table 279: Output fieldbus variable USINT (A3 $_{hex}$) – Instance 1...255

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

Common Services

Table 280: Output fieldbus variable USINT (A3 $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.35 Output Fieldbus Variable USINT Extended 1 (A4 $_{hex}$)

The extension of the "Output Fieldbus Variable USINT" class enables the exchange of PLC output variable data. The instance scope of the "Output Fieldbus Variable USINT Extended 1" class covers the PLC output variable data from 256 to 510.

For WAGO-I/O-*PRO* or CoDeSys, that means the PLC addresses for output variables %QB2807...%QB3061.

Instance 0 (Class Attributes)

Table 281: Output Fieldbus variable USINT Extended 1 (A4 $_{hex}$) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	255 (0x00FF)

Instance 256...510 (Output variable 256 up to 510)

Table 282: Output Fieldbus Variable USINT Extended 1 (A4 $_{hex}$) – Instance 256...510

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Fb_Out_Var	USINT	Fieldbus output variable of SPS	0

Common Services

Table 283: Output Fieldbus Variable USINT Extended 1 (A4 $_{hex}$) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.36 Output Fieldbus Variable USINT Extended 2 (A5 $_{hex}$)

The extension of the "Output Fieldbus Variable USINT" class enables the exchange of PLC output variable data. The instance scope of the "Output Fieldbus Variable USINT Extended 2" class covers the PLC output variable data from 511 to 512.

For WAGO-I/O-*PRO* or CoDeSys, that means the PLC addresses for output variables %QB3062...%QB3063.

Instance 0 (Class Attributes)

Table 284: Output Fieldbus Variable USINT Extended 2 (A5 $_{hex}$) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	2 (0x0002)

Instance 511...512 (Output variable 511 up to 512)

Table 285: Output Fieldbus Variable USINT Extended 2 (A5 $_{hex}$) – Instance 511...512

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

Common Services

Tabelle 286: Output Fieldbus Variable USINT Extended 2 (A5 $_{hex}$) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E $_{hex}$	Ja	Ja	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.37 Input Fieldbus Variable UINT (A6 $_{hex}$)

This class allows the reading of data from a particular PLC input variable. For WAGO-I/O-*PRO* or CoDeSys, that means the PLC addresses for input variables %IW1276...%IW1530.

Instance 0 (Class Attributes)

Table 287: Input fieldbus variable UINT (A6 $_{hex}$) – Class

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

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

Table 288: Input fieldbus variable UINT (A6 $_{hex}$) – Instance 1...255

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

Common Services

Table 289: Input fieldbus variable UINT (A6 $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.38 Input Fieldbus Variable UINT Extended 1 (A7 $_{hex}$)

The extension of the "Input Fieldbus Variable UINT" class enables the reading of PLC input variable data. The instance scope of the "Input Fieldbus Variable UINT Extended 1" class covers the PLC input variable data from the PLC input variable 256.

For WAGO-I/O-*PRO* or CoDeSys, that means the PLC addresses for input variable %IW1531.

Instanz 0 (Class Attributes)

Table 290: Input Fieldbus Variable UINT Extended 1 (A7 $_{hex}$) – Class

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

Instanz 256 (Input variable 256)

Table 291: Input Fieldbus Variable UINT Extended 1 (A7 $_{hex}$) – Instance 256

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

Common Services

Table 292: Input Fieldbus Variable UINT Extended 1 (A7 $_{hex}$) – Common service

Service code	Service available		Service Name	Description
	Class	Instance		
0E $_{hex}$	Ja	Ja	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	Nein	Ja	Set_Attribute_Single	Modifies an attribute value

12.3.5.39 Output Fieldbus Variable UINT (A8 $_{hex}$)

The class enables the exchange of data from a particular PLC output variable. For WAGO-I/O-*PRO* or CoDeSys, that means the PLC addresses for output variables %QW1276...%QW1530.

Instance 0 (Class Attributes)

Table 293: Output fieldbus variable UINT (A8 $_{hex}$) – Class

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

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

Table 294: Output fieldbus variable UINT (A8 $_{hex}$) – Instance 1...255

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

Common Services

Table 295: Output fieldbus variable UINT (A8 $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.40 Output Fieldbus Variable UINT Extended 1 (A9 $_{hex}$)

The extension of the "Output Fieldbus Variable UINT" class enables the exchange of PLC output variable data. The instance scope of the "Output Fieldbus Variable UINT Extended 1" class covers the PLC output variable data from PLC output variables 256.

For WAGO-I/O-*PRO* or CoDeSys, that means the PLC addresses for output variable %QW1531.

Instance 0 (Class Attributes)

Table 296: Output Fieldbus Variable UINT Extended 1 (A9 $_{hex}$) – Class

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

Instance 256 (Output variable 256)

Table 297: Output Fieldbus Variable UINT Extended 1 (A9 $_{hex}$) – Instance 256

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

Common Services

Table 298: Output Fieldbus Variable UINT Extended 1 (A9 $_{hex}$) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E $_{hex}$	Ja	Ja	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.41 Input Fieldbus Variable UDINT (AA $_{hex}$)

This class allows the reading of data from a particular PLC input variable. For WAGO-I/O-*PRO* or CoDeSys, that means the PLC addresses for input variables %ID638 ... %ID765.

Instance 0 (Class Attributes)

Table 299: Input fieldbus variable UDINT (AA $_{hex}$) – Class

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

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

Table 300: Input fieldbus variable UDINT (AA $_{hex}$) – Instance 1...128

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

Common Services

Table 301: Input fieldbus variable UDINT (AA $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.42 Input Fieldbus Variable UDINT Offset (AB $_{hex}$)

This class allows the reading of data from a particular PLC input variable.
With an offset of 2 bytes to the addresses of the "Input Fieldbus Variable UDINT (AA hex)" class, that means for WAGO-I/O-*PRO* or CoDeSys the PLC addresses for the input variables %ID638 ... %ID765.

Information

Information about Using the Offset
"Offset of 2 bytes" means:
If instance 1 of this class is read, you obtain High-Word of the address %ID638 and the Low-Word of the address %ID639, etc.
If instance 128 is read, you obtain only the High-Word of the address %ID765.

Instance 0 (Class Attributes)

Table 302: Input Fieldbus Variable UDINT Offset (AB $_{hex}$) – Class

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

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

Table 303: Input Fieldbus Variable UDINT Offset (AB $_{hex}$) – Instance 1...128

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

Common Services

Table 304: Input Fieldbus Variable UDINT Offset (AB $_{hex}$) – Common service

Servicecode	Service available		Service-Name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute
10 $_{hex}$	No	Yes	Set_Attribute_Single	Modifies an attribute value

12.3.5.43 Output Fieldbus Variable UDINT (AC $_{hex}$)

The class enables the exchange of data from a particular PLC output variable. For WAGO-I/O-PRO or CoDeSys, that means the PLC addresses for output variables %QD638...%QD765.

Instance 0 (Class Attributes)

Table 305: Input fieldbus variable UDINT (AA $_{hex}$) – Class

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

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

Table 306: Input fieldbus variable UDINT (AA $_{hex}$) – Instance 1...128

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

Common Services

Table 307: Input fieldbus variable UDINT (AA $_{hex}$) – Common service

Service code	Service available		Service name	Description
	Class	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

12.3.5.44 Output Fieldbus Variable UDINT Offset (AD $_{hex}$)

The class enables the exchange of data from a particular PLC output variable. With an offset of 2 bytes to the addresses of the "Output Fieldbus Variable UDINT (AC $_{hex}$)" class, that means for WAGO-I/O-*PRO* or CoDeSys the PLC addresses for %QD638 ... %QD765.

Information

Information about Using the Offset

"Offset of 2 bytes" means:
If instance 1 of this class is read, you obtain High-Word of the address %ID638 and the Low-Word of the address %ID639, etc.
If instance 128 is read, you obtain only the High-Word of the address %ID765..

Instance 0 (Class Attributes)

Table 308: Output Fieldbus Variable UDINT Offset (AD $_{hex}$) – Class

Attribute ID	Access	Name	Data type	Description	Default value
1	Get	Revision	UINT	Revision of this object	1 (0x0001)
2	Get	Max Instance	UINT	Max. number of instances	128 (0x0080)

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

Table 309: Output Fieldbus Variable UDINT Offset (AD $_{hex}$) – Instance 1...128

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

Common Services

Table 310: Output Fieldbus Variable UDINT Offset (AD $_{hex}$) – Common service

Servicecode	Service available		Service-Name	Description
	Casse	Instance		
0E $_{hex}$	Yes	Yes	Get_Attribute_Single	Supplies contents of the appropriate attribute

13 I/O Modules

13.1 Overview

For modular applications with the WAGO-I/O-SYSTEM 750, different types of I/O modules are available

- Digital Input Modules
- Digital Output Modules
- Analog Input Modules
- Analog Output Modules
- Special Modules
- System Modules

For detailed information on the I/O modules and the module variations, refer to the manuals for the I/O modules.

You will find these manuals on DVD ROM "AUTOMATION Tools and Docs" (order no. 0888-0412) or on the WAGO web pages under http://www.wago.com → Service → Download → Documentation.

Information

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

13.2 Process Data Architecture for MODBUS/TCP

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

MODBUS/TCP process image uses a word structure (with word alignment). The internal mapping method for data greater than one byte conforms to the Intel format.

The following section describes the process image for various WAGO-I/O-SYSTEM 750 and 753 I/O modules with MODBUS/TCP.

NOTICE

Equipment damage due to incorrect address!
Depending on the specific position of an I/O module in the fieldbus node, the process data of all previous byte or bit-oriented modules must be taken into account to determine its location in the process data map.

The structure of the process data mapping is identical for the PFC process image of the programmable fieldbus controller.

13.2.1 Digital Input Modules

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

Some digital modules have an additional diagnostic bit per channel in the Input Process Image. The diagnostic bit is used for detecting faults that occur (e.g., wire breaks and/or short circuits).

When analog input modules are also present in the node, the digital data is always appended after the analog data in the Input Process Image, grouped into bytes.

13.2.1.1 1 Channel Digital Input Module with Diagnostics

750-435

Table 311: 1 Channel Digital Input Module with Diagnostics

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
						Diagnostic bit S 1	Data bit DI 1

13.2.1.2 2 Channel Digital Input Modules

750-400, -401, -405, -406, -410, -411, -412, -427, -438, (and all variations),
753-400, -401, -405, -406, -410, -411, -412, -427

Table 312: 2 Channel Digital Input Modules

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
						Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.3 2 Channel Digital Input Module with Diagnostics

750-419, -421, -424, -425,
753-421, -424, -425

Table 313: 2 Channel Digital Input Module with Diagnostics

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.4 2 Channel Digital Input Module with Diagnostics and Output Process Data

750-418,
753-418

The digital input module supplies a diagnostic and acknowledge bit for each input channel. If a fault condition occurs, the diagnostic bit is set. After the fault condition is cleared, an acknowledge bit must be set to re-activate the input. The diagnostic data and input data bit is mapped in the Input Process Image, while the acknowledge bit is in the Output Process Image.

Table 314: 2 Channel Digital Input Module with Diagnostics and Output Process Data

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

Output Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				Acknowledge-ment bit Q 2 Channel 2	Acknowledge-ment bit Q 1 Channel 1	0	0

13.2.1.5 4 Channel Digital Input Modules

750-402, -403, -408, -409, -414, -415, -422, -423, -428, -432, -433, -1420, -1421, -1422
753-402, -403, -408, -409, -415, -422, -423, -428, -432, -433, -440

Table 315: 4 Channel Digital Input Modules

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.6 8 Channel Digital Input Modules

750-430, -431, -436, -437, -1415, -1416, -1417
753-430, -431, -434

Table 316: 8 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.1.7 16 Channel Digital Input Modules

750-1400, -1402, -1405, -1406, -1407

Table 317: 16 Channel Digital Input Modules

Input Process Image															
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 16 Channel 16	Data bit DI 15 Channel 15	Data bit DI 14 Channel 14	Data bit DI 13 Channel 13	Data bit DI 12 Channel 12	Data bit DI 11 Channel 11	Data bit DI 10 Channel 10	Data bit DI 9 Channel 9	Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.2.2 Digital Output Modules

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

Some digital modules have an additional diagnostic bit per channel in the Input Process Image. The diagnostic bit is used for detecting faults that occur (e.g., wire breaks and/or short circuits). For modules with diagnostic bit is set, also the data bits have to be evaluated.

When analog output modules are also present in the node, the digital image data is always appended after the analog data in the Output Process Image, grouped into bytes.

13.2.2.1 1 Channel Digital Output Module with Input Process Data

750-523

The digital output modules deliver 1 bit via a process value Bit in the output process image, which is illustrated in the input process image. This status image shows "manual mode".

Table 318: 1 Channel Digital Output Module with Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						not used	Status bit “Manual Operation“

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

13.2.2.2 2 Channel Digital Output Modules

750-501, -502, -509, -512, -513, -514, -517, -535, (and all variations),
753-501, -502, -509, -512, -513, -514, -517

Table 319: 2 Channel Digital Output Modules

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

13.2.2.3 2 Channel Digital Input Modules with Diagnostics and Input Process Data

750-507 (-508), -522,
753-507

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 320: 2 Channel Digital Input Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

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

750-506,
753-506

The digital output module has 2-bits of diagnostic information for each output channel. The 2-bit diagnostic information can then be decoded to determine the exact fault condition of the module (i.e., overload, a short circuit, or a broken wire). The 4-bits of diagnostic data are mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 321: 2 Channel Digital Input Modules with Diagnostics and Input Process Data 75x-506

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 3 Channel 2	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Diagnostic bit S 0 Channel 1

Diagnostic bits S1/S0, S3/S2: = ‘00’	standard mode
Diagnostic bits S1/S0, S3/S2: = ‘01’	no connected load/short circuit against +24 V
Diagnostic bits S1/S0, S3/S2: = ‘10’	Short circuit to ground/overload

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

13.2.2.4 4 Channel Digital Output Modules

750-504, -516, -519, -531,
753-504, -516, -531, -540

Table 322: 4 Channel Digital Output Modules

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

13.2.2.5 4 Channel Digital Output Modules with Diagnostics and Input Process Data

750-532

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 323: 4 Channel Digital Output Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 4 Channel 4	Diagnostic bit S 3 Channel 3	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Diagnostic bit S = ‘0’ no Error
Diagnostic bit S = ‘1’ overload, short circuit, or broken wire

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

13.2.2.6 8 Channel Digital Output Module

750-530, -536, -1515, -1516
753-530, -534

Table 324: 8 Channel Digital Output Module

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

13.2.2.7 8 Channel Digital Output Modules with Diagnostics and Input Process Data

750-537

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 325: 8 Channel Digital Output Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Diagnostic bit S 8 Channel 8	Diagnostic bit S 7 Channel 7	Diagnostic bit S 6 Channel 6	Diagnostic bit S 5 Channel 5	Diagnostic bit S 4 Channel 4	Diagnostic bit S 3 Channel 3	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Diagnostic bit S = ‘0’ no Error
Diagnostic bit S = ‘1’ overload, short circuit, or broken wire

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

13.2.2.8 16 Channel Digital Output Modules

750-1500, -1501, -1504, -1505

Table 326: 16 Channel Digital Output Modules

Output Process Image															
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 16 Channel 16	controls DO 15 Channel 15	controls DO 14 Channel 14	controls DO 13 Channel 13	controls DO 12 Channel 12	controls DO 11 Channel 11	controls DO 10 Channel 10	controls DO 9 Channel 9	controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

13.2.2.9 8 Channel Digital Input/Output Modules

750-1502, -1506

Table 327: 8 Channel Digital Input/Output Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

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

13.2.3 Analog Input Modules

The hardware of an analog input module has 16 bits of measured analog data per channel and 8 bits of control/status.

However, the coupler/controller with MODBUS/TCP does not have access to the 8 control/status bits.

Therefore, the coupler/controller with MODBUS/TCP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Input Process Image.

When digital input modules are also present in the node, the analog input data is always mapped into the Input Process Image in front of the digital data.

Information

Information to the structure of the Control/Status byte
For detailed information about the structure of a particular module's control/status byte, please refer to that module's manual. Manuals for each module can be found on the Internet under: http://www.wago.com.

13.2.3.1 1 Channel Analog Input Modules

750-491, (and all variations)

Table 328: 1 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination High Byte	Byte Destination Low Byte	Description
0	D1	D0	Measured Value U_D
1	D3	D2	Measured Value U_{ref}

13.2.3.2 2 Channel Analog Input Modules

750-452, -454, -456, -461, -462, -465, -466, -467, -469, -472, -474, -475, 476, -477, -478, -479, -480, -481, -483, -485, -492, (and all variations),
753-452, -454, -456, -461, -465, -466, -467, -469, -472, -474, -475, 476, -477, 478, -479, -483, -492, (and all variations)

Table 329: 2 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination High Byte	Byte Destination Low Byte	Description
0	D1	D0	Measured Value Channel 1
1	D3	D2	Measured Value Channel 2

13.2.3.3 4 Channel Analog Input Modules

750-453, -455, -457, -459, -460, -468, (and all variations),
753-453, -455, -457, -459

Table 330: 4 Channel Analog Input Modules

Input Process Image			
Offset	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
0	D1	D0	Measured Value Channel 1
1	D3	D2	Measured Value Channel 2
2	D5	D4	Measured Value Channel 3
3	D7	D6	Measured Value Channel 4

13.2.4 Analog Output Modules

The hardware of an analog output module has 16 bits of measured analog data per channel and 8 bits of control/status. However, the coupler/controller with MODBUS/TCP does not have access to the 8 control/status bits. Therefore, the coupler/controller with MODBUS/TCP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Output Process Image.

When digital output modules are also present in the node, the analog output data is always mapped into the Output Process Image in front of the digital data.

Information

Information to the structure of the Control/Status byte
For detailed information about the structure of a particular module’s control/status byte, please refer to that module’s manual. Manuals for each module can be found on the Internet under: http://www.wago.com.

13.2.4.1 2 Channel Analog Output Modules

750-550, -552, -554, -556, -560, -562, 563, -585, (and all variations),
753-550, -552, -554, -556

Table 331: 2 Channel Analog Output Modules

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Output Value Channel 1
1	D3	D2	Output Value Channel 2

13.2.4.2 4 Channel Analog Output Modules

750-553, -555, -557, -559,
753-553, -555, -557, -559

Table 332: 4 Channel Analog Output Modules

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Output Value Channel 1
1	D3	D2	Output Value Channel 2
2	D5	D4	Output Value Channel 3
3	D7	D6	Output Value Channel 4

13.2.5 Specialty Modules

WAGO has a host of Specialty I/O modules that perform various functions. With individual modules beside the data bytes also the control/status byte is mapped in the process image.

The control/status byte is required for the bidirectional data exchange of the module with the higher-ranking control system. The control byte is transmitted from the control system to the module and the status byte from the module to the control system.
This allows, for example, setting of a counter with the control byte or displaying of overshooting or undershooting of the range with the status byte.

The control/status byte always is in the process image in the Low byte.

Information

Information to the structure of the Control/Status byte
For detailed information about the structure of a particular module's control/status byte, please refer to that module's manual. Manuals for each module can be found on the Internet under: http://www.wago.com.

13.2.5.1 Counter Modules

750-404, (and all variations except of /000-005),
753-404, (and variation /000-003)

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/status). The counter value is supplied as 32 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

Table 333: Counter Modules 750-404, (and all variations except of /000-005), 753-404, (and variation /000-003)

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	S	Status byte
1	D1	D0	Counter value
2	D3	D2	

Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C	Control byte
1	D1	D0	Counter setting value
2	D3	D2	

750-404/000-005

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/ status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

Table 334: Counter Modules 750-404/000-005

Input Process Image			
Offset	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
0	-	S	Status byte
1	D1	D0	Counter Value of Counter 1
2	D3	D2	Counter Value of Counter 2

Output Process Image			
Offset	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
0	-	C	Control byte
1	D1	D0	Counter Setting Value of Counter 1
2	D3	D2	Counter Setting Value of Counter 2

750-638,
753-638

The above Counter Modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 2 bytes of control/status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

Table 335: Counter Modules 750-638, 753-638

Input Process Image			
Offset	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
0	-	S0	Status byte von Counter 1
1	D1	D0	Counter Value von Counter 1
2	-	S1	Status byte von Counter 2
3	D3	D2	Counter Value von Counter 2

Output Process Image			
Offset	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
0	-	C0	Control byte von Counter 1
1	D1	D0	Counter Setting Value von Counter 1
2	-	C1	Control byte von Counter 2
3	D3	D2	Counter Setting Value von Counter 2

13.2.5.2 Pulse Width Modules

750-511, (and all variations /xxx-xxx)

The above Pulse Width modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of channel data and 2 bytes of control/status). The two channel values are supplied as 16 bits. Each channel has its own control/status byte. The following table illustrates the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

Table 336: Pulse Width Modules 750-511, /xxx-xxx

Input and Output Process			
Offset	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
0	-	C0/S0	Control/Status byte of Channel 1
1	D1	D0	Data Value of Channel 1
2	-	C1/S1	Control/Status byte of Channel 2
3	D3	D2	Data Value of Channel 2

13.2.5.3 Serial Interface Modules with alternative Data Format

750-650, (and the variations /000-002, -004, -006, -009, -010, -011, -012, -013),
750-651, (and the variations /000-001, -002, -003),
750-653, (and the variations /000-002, -007),
753-650, -653

Note

The process image of the / 003-000-variants depends on the parameterized operating mode!
With the freely parametrizable variations /003 000 of the serial interface modules, the desired operation mode can be set. Dependent on it, the process image of these modules is then the same, as from the appropriate variation.

The above Serial Interface Modules with alternative data format have a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 2 words mapped into each image. Word alignment is applied.

Table 337: Serial Interface Modules with alternative Data Format

Input and Output Process Image				
Offset	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
0	D0	C/S	Data byte	Control/status byte
1	D2	D1	Data bytes	

13.2.5.4 Serial Interface Modules with Standard Data Format

750-650/000-001, -014, -015, -016
750-653/000-001, -006

The above Serial Interface Modules with Standard Data Format have a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 3 words mapped into each image. Word alignment is applied.

Table 338: Serial Interface Modules with Standard Data Format

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	C/S	Data byte	Control/status byte
1	D2	D1	Data bytes	
2	D4	D3		

13.2.5.5 Data Exchange Module

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

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

Table 339: Data Exchange Module

Input and Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Data bytes
1	D3	D2	

13.2.5.6 SSI Transmitter Interface Modules

750-630 (and all variations)

Note

The process image of the / 003-000-variants depends on the parameterized operating mode!
The operating mode of the configurable /003-000 I/O module versions can be set. Based on the operating mode, the process image of these I/O modules is then the same as that of the respective version.

The above SSI Transmitter Interface modules have a total of 4 bytes of user data in the Input Process Image, which has 2 words mapped into the image.
Word alignment is applied.

Table 340: SSI Transmitter Interface Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Data bytes
1	D3	D2	

13.2.5.7 Incremental Encoder Interface Modules

750-631/000-004, -010, -011

The above Incremental Encoder Interface modules have 5 bytes of input data and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which have 4 words into each image. Word alignment is applied.

Table 341: Incremental Encoder Interface Modules 750-631/000-004, --010, -011

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	S	not used	Status byte
1	D1	D0	Counter word	
2	-	-	not used	
3	D4	D3	Latch word	

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C	not used	Control byte
1	D1	D0	Counter setting word	
2	-	-	not used	
3	-	-	not used	

750-634

The above Incremental Encoder Interface module has 5 bytes of input data (6 bytes in cycle duration measurement mode) and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which has 4 words mapped into each image. Word alignment is applied.

Table 342: Incremental Encoder Interface Modules 750-634

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	S	not used	Status byte
1	D1	D0	Counter word	
2	-	(D2) *)	not used	(Periodic time)
3	D4	D3	Latch word	

*) If cycle duration measurement mode is enabled in the control byte, the cycle duration is given as a 24-bit value that is stored in D2 together with D3/D4.

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C	not used	Control byte
1	D1	D0	Counter setting word	
2	-	-	not used	
3	-	-		

750-637

The above Incremental Encoder Interface Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of encoder data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Table 343: Incremental Encoder Interface Modules 750-637

Input and Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	-	C0/S0	Control/Status byte of Channel 1
1	D1	D0	Data Value of Channel 1
2	-	C1/S1	Control/Status byte of Channel 2
3	D3	D2	Data Value of Channel 2

750-635,
753-635

The above Digital Pulse Interface module has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

Table 344: Digital Pulse Interface Modules 750-635

Input and Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	D0	C0/S0	Data byte	Control/status byte
1	D2	D1	Data bytes	

13.2.5.8 DC-Drive Controller

750-636

The DC-Drive Controller maps 6 bytes into both the input and output process image. The data sent and received are stored in up to 4 input and output bytes (D0 ... D3). Two control bytes (C0, C1) and two status bytes (S0/S1) are used to control the I/O module and the drive.

In addition to the position data in the input process image (D0 … D3), it is possible to display extended status information (S2 … S5). Then the three control bytes (C1 … C3) and status bytes (S1 … S3) are used to control the data flow.

Bit 3 of control byte C1 (C1.3) is used to switch between the process data and the extended status bytes in the input process image (Extended Info_ON). Bit 3 of status byte S1 (S1.3) is used to acknowledge the switching process.

Table 345: DC-Drive Controller 750-636

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	S1	S0	Status byte S1	Status byte S0
1	D1*) / S3**)	D0*) / S2**)	Actual position*) / Extended status byte S3**)	Actual position (LSB) / Extended status byte S2**)
2	D3*) / S5**)	D2*) / S4**)	Actual position (MSB) / Extended status byte S3**)	Actual position*) / Extended status byte S4**)

*) ExtendedInfo_ON = ‘0’.
**) ExtendedInfo_ON = ‘1’.

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	C1	C0	Control byte C1	Control byte C0
1	D1	D0	Setpoint position	Setpoint position (LSB)
2	D3	D2	Setpoint position (MSB)	Setpoint position

13.2.5.9 Stepper Controller

750-670

The Stepper controller RS422 / 24 V / 20 mA 750-670 provides the fieldbus coupler 12 bytes input and output process image via 1 logical channel. The data to be sent and received are stored in up to 7 output bytes (D0 … D6) and 7 input bytes (D0 … D6), depending on the operating mode.

Output byte D0 and input byte D0 are reserved and have no function assigned.

One I/O module control and status byte (C0, S0) and 3 application control and status bytes (C1 ... C3, S1 ... S3) provide the control of the data flow.

Switching between the two process images is conducted through bit 5 in the control byte (C0 (C0.5). Activation of the mailbox is acknowledged by bit 5 of the status byte S0 (S0.5).

Table 346: Stepper Controller RS 422 / 24 V / 20 mA 750-670

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	reserved	S0	reserved	Status byte S0
1	D1	D0	Process data*) / Mailbox**)	
2	D3	D2		
3	D5	D4		
4	S3	D6	Status byte S3	Process data*) / reserved**)
5	S1	S2	Status byte S1	Status byte S2

*) Cyclic process image (Mailbox disabled)
**) Mailbox process image (Mailbox activated)

Output Process Image				
Offset	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
0	reserved	C0	reserved	Control byte C0
1	D1	D0	Process data*) / Mailbox**)	
2	D3	D2		
3	D5	D4		
4	C3	D6	Control byte C3	Process data*) / reserved**)
5	C1	C2	Control byte C1	Control byte C2

*) Cyclic process image (Mailbox disabled)
**) Mailbox process image (Mailbox activated)

13.2.5.10 RTC Module

750-640

The RTC Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of module data and 1 byte of control/status and 1 byte ID for command). The following table illustrates the Input and Output Process Image, which have 3 words mapped into each image. Word alignment is applied.

Table 347: RTC Module 750-640

Input and Output Process Image				
Offset	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
0	ID	C/S	Command byte	Control/status byte
1	D1	D0	Data bytes	
2	D3	D2		

13.2.5.11 DALI/DSI Master Module

750-641

The DALI/DSI Master module has a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of module data and 1 byte of control/status). The following tables illustrate the Input and Output Process Image, which have 3 words mapped into each image. Word alignment is applied.

Table 348: DALI/DSI Master module 750-641

Input Process Image				
Offset	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
0	D0	S	DALI Response	Status byte
1	D2	D1	Message 3	DALI Address
2	D4	D3	Message 1	Message 2

Output Process Image				
Offset	Byte Destination High Byte	Byte Destination Low Byte	Description	
0	D0	C	DALI command, DSI dimming value	Control byte
1	D2	D1	Parameter 2	DALI Address
2	D4	D3	Command extension	Parameter 1

13.2.5.12 EnOcean Radio Receiver

750-642

The EnOcean radio receiver has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following tables illustrate the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

Table 349: EnOcean Radio Receiver 750-642

Input Process Image				
Offset	Byte Destination High Byte	Byte Destination Low Byte	Description	
0	D0	S	Data byte	Status byte
1	D2	D1	Data bytes	

Output Process Image				
Offset	Byte Destination High Byte	Byte Destination Low Byte	Description	
0	-	C	not used	Control byte
1	-	-	not used	

13.2.5.13 MP Bus Master Module

750-643

The MP Bus Master Module has a total of 8 bytes of user data in both the Input and Output Process Image (6 bytes of module data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Table 350: MP Bus Master Module 750-643

Input and Output Process Image				
Offset	Byte Destination High Byte	Byte Destination Low Byte	Description	
0	C1/S1	C0/S0	extended Control/ Status byte	Control/status byte
1	D1	D0	Data bytes	
2	D3	D2		
3	D5	D4		

13.2.5.14 *Bluetooth*® RF-Transceiver

750-644

The size of the process image for the *Bluetooth*® module can be adjusted to 12, 24 or 48 bytes.
It consists of a control byte (input) or status byte (output); an empty byte; an overlayable mailbox with a size of 6, 12 or 18 bytes (mode 2); and the *Bluetooth*® process data with a size of 4 to 46 bytes.
Thus, each *Bluetooth*® module uses between 12 and 48 bytes in the process image. The sizes of the input and output process images are always the same.

The first byte contains the control/status byte; the second contains an empty byte.

Process data attach to this directly when the mailbox is hidden. When the mailbox is visible, the first 6, 12 or 18 bytes of process data are overlaid by the mailbox data, depending on their size. Bytes in the area behind the optionally visible mailbox contain basic process data. The internal structure of the *Bluetooth*® process data can be found in the documentation for the *Bluetooth*® 750-644 RF Transceiver.

The mailbox and the process image sizes are set with the startup tool WAGO-I/O-*CHECK*.

Table 351: Bluetooth® RF-Transceiver 750-644

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

13.2.5.15 Vibration Velocity/Bearing Condition Monitoring VIB I/O

750-645

The Vibration Velocity/Bearing Condition Monitoring VIB I/O has a total of 12 bytes of user data in both the Input and Output Process Image (8 bytes of module data and 4 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 8 words mapped into each image.
Word alignment is applied.

Table 352: Vibration Velocity/Bearing Condition Monitoring VIB I/O 750-645

<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 rowspan="2" colspan="2">Description</th></tr>
<tr><th>High Byte</th><th>Low Byte</th></tr>
<tr><td>0</td><td>-</td><td>C0/S0</td><td>not used</td><td>Control/status byte (log. Channel 1, Sensor input 1)</td></tr>
<tr><td>1</td><td>D1</td><td>D0</td><td colspan="2">Data bytes (log. Channel 1, Sensor input 1)</td></tr>
<tr><td>2</td><td>-</td><td>C1/S1</td><td>not used</td><td>Control/status byte (log. Channel 2, Sensor input 2)</td></tr>
<tr><td>3</td><td>D3</td><td>D2</td><td colspan="2">Data bytes (log. Channel 2, Sensor input 2)</td></tr>
<tr><td>4</td><td>-</td><td>C2/S2</td><td>not used</td><td>Control/status byte (log. Channel 3, Sensor input 1)</td></tr>
<tr><td>5</td><td>D5</td><td>D4</td><td colspan="2">Data bytes (log. Channel 3, Sensor input 3)</td></tr>
<tr><td>6</td><td>-</td><td>C3/S3</td><td>not used</td><td>Control/status byte (log. Channel 4, Sensor input 2)</td></tr>
<tr><td>7</td><td>D7</td><td>D6</td><td colspan="2">Data bytes (log. Channel 4, Sensor input 2)</td></tr>
</table>

13.2.5.16 KNX/EIB/TP1 Module

753-646

The KNX/TP1 module appears in router and device mode with a total of 24-byte user data within the input and output area of the process image, 20 data bytes and 2 control/status bytes. Even though the additional bytes S1 or C1 are transferred as data bytes, they are used as extended status and control bytes. The opcode is used for the read/write command of data and the triggering of specific functions of the KNX/EIB/TP1 module. Word-alignment is used to assign 12 words in the process image. Access to the process image is not possible in router mode. Telegrams can only be tunneled.

In device mode, access to the KNX data can only be performed via special function blocks of the IEC application. Configuration using the ETS engineering tool software is required for KNX.

Table 353: KNX/EIB/TP1 Module 753-646

Input Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	S0	not used	Status byte
1	S1	OP	extended Status byte	Opcode
2	D1	D0	Data byte 1	Data byte 0
3	D3	D2	Data byte 3	Data byte 2
4	D5	D4	Data byte 5	Data byte 4
5	D7	D6	Data byte 7	Data byte 6
6	D9	D8	Data byte 9	Data byte 8
7	D11	D10	Data byte 11	Data byte 10
8	D13	D12	Data byte 13	Data byte 12
9	D15	D14	Data byte 15	Data byte 14
10	D17	D16	Data byte 17	Data byte 16
11	D19	D18	Data byte 19	Data byte 18

Output Process Image				
Offset	Byte Destination		Description	
	High Byte	Low Byte		
0	-	C0	not used	Control byte
1	C1	OP	extended Control byte	Opcode
2	D1	D0	Data byte 1	Data byte 0
3	D3	D2	Data byte 3	Data byte 2
4	D5	D4	Data byte 5	Data byte 4
5	D7	D6	Data byte 7	Data byte 6
6	D9	D8	Data byte 9	Data byte 8
7	D11	D10	Data byte 11	Data byte 10
8	D13	D12	Data byte 13	Data byte 12
9	D15	D14	Data byte 15	Data byte 14
10	D17	D16	Data byte 17	Data byte 16
11	D19	D18	Data byte 19	Data byte 18

13.2.5.17 AS-interface Master Module

750-655

The length of the process image of the AS-interface master module can be set to fixed sizes of 12, 20, 24, 32, 40 or 48 bytes.
It consists of a control or status byte, a mailbox with a size of 0, 6, 10, 12 or 18 bytes and the AS-interface process data, which can range from 0 to 32 bytes.

The AS-interface master module has a total of 6 to maximally 24 words data in both the Input and Output Process Image. Word alignment is applied.

The first Input and output word, which is assigned to an AS-interface master module, contains the status / control byte and one empty byte.
Subsequently the mailbox data are mapped, when the mailbox is permanently superimposed (Mode 1).

In the operating mode with suppressible mailbox (Mode 2), the mailbox and the cyclical process data are mapped next.
The following words contain the remaining process dat.

The mailbox and the process image sizes are set with the startup tool WAGO-I/O-*CHECK*.

Table 354: AS-interface Master module 750-655

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

13.2.6 System Modules

13.2.6.1 System Modules with Diagnostics

750-610, -611

The modules provide 2 bits of diagnostics in the Input Process Image for monitoring of the internal power supply.

Table 355: System Modules with Diagnostics 750-610, -611

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
						Diagnostic bit S 2 Fuse	Diagnostic bit S 1 Fuse

13.2.6.2 Binary Space Module

750-622

The Binary Space Modules behave alternatively like 2 channel digital input modules or output modules and seize depending upon the selected settings 1, 2, 3 or 4 bits per channel. According to this, 2, 4, 6 or 8 bits are occupied then either in the process input or the process output image.

Table 356: Binary Space Module 750-622 (with behavior like 2 channel digital input)

Input and Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(Data bit DI 8)	(Data bit DI 7)	(Data bit DI 6)	(Data bit DI 5)	(Data bit DI 4)	(Data bit DI 3)	Data bit DI 2	Data bit DI 1

13.3 Process Data Architecture for EtherNet/IP

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

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

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

For the PFC process image of the programmable fieldbus controller is the structure of the process data mapping identical.

NOTICE

Equipment damage due to incorrect address!
Depending on the specific position of an I/O module in the fieldbus node, the process data of all previous byte or bit-oriented modules must be taken into account to determine its location in the process data map.

13.3.1 Digital Input Modules

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

Some digital I/O modules have an additional diagnostic bit per channel in the input process image. The diagnostic bit detects faults (e.g., wire breakage, overloads and/or short circuits). For some I/O modules, the data bits also have be evaluated with the set diagnostic bit.

When analog input modules are also present in the node, the digital data is always appended after the analog data in the Input Process Image, grouped into bytes.

1 sub index is assigned for each 8 bit.

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

13.3.1.1 1 Channel Digital Input Module with Diagnostics

750-435

Table 357: 1 Channel Digital Input Module with Diagnostics

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
						Diagnostic bit S 1	Data bit DI 1

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

13.3.1.2 2 Channel Digital Input Modules

750-400, -401, -405, -406, -410, -411, -412, -427, -438, (and all variations),
753-400, -401, -405, -406, -410, -411, -412, -427

Table 358: 2 Channel Digital Input Modules

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
						Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

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

13.3.1.3 2 Channel Digital Input Module with Diagnostics

750-419, -421, -424, -425,
753-421, -424, -425

Table 359: 2 Channel Digital Input Module with Diagnostics

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

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

13.3.1.4 2 Channel Digital Input Module with Diagnostics and Output Process Data

750-418,
753-418

The digital input module supplies a diagnostic and acknowledge bit for each input channel. If a fault condition occurs, the diagnostic bit is set. After the fault condition is cleared, an acknowledge bit must be set to re-activate the input. The diagnostic data and input data bit is mapped in the Input Process Image, while the acknowledge bit is in the Output Process Image.

Table 360: 2 Channel Digital Input Module with Diagnostics and Output Process Data

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

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

Output Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				Acknowledge-ment bit Q 2 Channel 2	Acknowledge-ment bit Q 1 Channel 1	0	0

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

13.3.1.5 4 Channel Digital Input Modules

750-402, -403, -408, -409, -414, -415, -422, -423, -428, -432, -433, -1420, -1421, -1422
753-402, -403, -408, -409, -415, -422, -423, -428, -432, -433, -440

Table 361: 4 Channel Digital Input Modules

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
				Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

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

13.3.1.6 8 Channel Digital Input Modules

750-430, -431, -436, -437, -1415, -1416, -1417
753-430, -431, -434

Table 362: 8 Channel Digital Input Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

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

13.3.1.7 16 Channel Digital Input Modules

750-1400, -1402, -1405, -1406, -1407

Table 363: 16 Channel Digital Input Modules

Input Process Image															
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 16 Channel 16	Data bit DI 15 Channel 15	Data bit DI 14 Channel 14	Data bit DI 13 Channel 13	Data bit DI 12 Channel 12	Data bit DI 11 Channel 11	Data bit DI 10 Channel 10	Data bit DI 9 Channel 9	Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

13.3.2 Digital Output Modules

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

Some digital modules have an additional diagnostic bit per channel in the Input Process Image. The diagnostic bit is used for detecting faults that occur (e.g., wire breaks and/or short circuits). With some I/O modules, with set diagnostic bit, additionally the data bits must be evaluated.

When analog output modules are also present in the node, the digital image data is always appended after the analog data in the Output Process Image, grouped into bytes.

For each 8 bits a subindex is occupied.

Each output channel occupies one instance in the Discrete Output Point Object (Class 0x 66).

13.3.2.1 1 Channel Digital Output Module with Input Process Data

750-523

The digital output modules deliver 1 bit via a process value Bit in the output process image, which is illustrated in the input process image. This status image shows "manual mode".

Table 364: 1 Channel Digital Output Module with Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						not used	Status bit “Manual Operation“

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

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

13.3.2.2 2 Channel Digital Output Modules

750-501, -502, -509, -512, -513, -514, -517, -535, (and all variations),
753-501, -502, -509, -512, -513, -514, -517

Table 365: 2 Channel Digital Output Modules

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

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

13.3.2.3 2 Channel Digital Input Modules with Diagnostics and Input Process Data

750-507 (-508), -522,
753-507

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 366: 2 Channel Digital Input Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

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

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

750-506,
753-506

The digital output module has 2-bits of diagnostic information for each output channel. The 2-bit diagnostic information can then be decoded to determine the exact fault condition of the module (i.e., overload, a short circuit, or a broken wire). The 4-bits of diagnostic data are mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 367: 2 Channel Digital Input Modules with Diagnostics and Input Process Data 75x-506

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 3 Channel 2	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1	Diagnostic bit S 0 Channel 1

Diagnostic bits S1/S0, S3/S2: = ‘00’ standard mode
Diagnostic bits S1/S0, S3/S2: = ‘01’ no connected load/short circuit against +24 V
Diagnostic bits S1/S0, S3/S2: = ‘10’ Short circuit to ground/overload

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

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

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

13.3.2.4 4 Channel Digital Output Modules

750-504, -516, -519, -531,
753-504, -516, -531, -540

Table 368: 4 Channel Digital Output Modules

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

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

13.3.2.5 4 Channel Digital Output Modules with Diagnostics and Input Process Data

750-532

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 369: 4 Channel Digital Output Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				Diagnostic bit S 4 Channel 4	Diagnostic bit S 3 Channel 3	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Diagnostic bit S = ‘0’ no Error
Diagnostic bit S = ‘1’ overload, short circuit, or broken wire

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

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

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

13.3.2.6 8 Channel Digital Output Module

750-530, -536, -1515, -1516
753-530, -534

Table 370: 8 Channel Digital Output Module

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

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

13.3.2.7 8 Channel Digital Output Modules with Diagnostics and Input Process Data

750-537

The digital output modules have a diagnostic bit for each output channel. When an output fault condition occurs (i.e., overload, short circuit, or broken wire), a diagnostic bit is set. The diagnostic data is mapped into the Input Process Image, while the output control bits are in the Output Process Image.

Table 371: 8 Channel Digital Output Modules with Diagnostics and Input Process Data

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Diagnostic bit S 8 Channel 8	Diagnostic bit S 7 Channel 7	Diagnostic bit S 6 Channel 6	Diagnostic bit S 5 Channel 5	Diagnostic bit S 4 Channel 4	Diagnostic bit S 3 Channel 3	Diagnostic bit S 2 Channel 2	Diagnostic bit S 1 Channel 1

Diagnostic bit S = ‘0’ no Error
Diagnostic bit S = ‘1’ overload, short circuit, or broken wire

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

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

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

13.3.2.8 16 Channel Digital Output Modules

750-1500, -1501, -1504, -1505

Table 372: 16 Channel Digital Output Modules

Output Process Image															
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
controls DO 16 Channel 16	controls DO 15 Channel 15	controls DO 14 Channel 14	controls DO 13 Channel 13	controls DO 12 Channel 12	controls DO 11 Channel 11	controls DO 10 Channel 10	controls DO 9 Channel 9	controls DO 8 Channel 8	controls DO 7 Channel 7	controls DO 6 Channel 6	controls DO 5 Channel 5	controls DO 4 Channel 4	controls DO 3 Channel 3	controls DO 2 Channel 2	controls DO 1 Channel 1

The output modules seize 16 Instances in Class (0x66).

13.3.2.9 8 Channel Digital Input/Output Modules

750-1502, -1506

Table 373: 8 Channel Digital Input/Output Modules

Input Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data bit DI 8 Channel 8	Data bit DI 7 Channel 7	Data bit DI 6 Channel 6	Data bit DI 5 Channel 5	Data bit DI 4 Channel 4	Data bit DI 3 Channel 3	Data bit DI 2 Channel 2	Data bit DI 1 Channel 1

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

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

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

13.3.3 Analog Input Modules

The hardware of an analog input module has 16 bits of measured analog data per channel and 8 bits of control/status.

However, the coupler/controller with EtherNet/IP does not have access to the 8 control/status bits.

Therefore, the coupler/controller with MODBUS/TCP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Input Process Image.

When digital input modules are also present in the node, the analog input data is always mapped into the Input Process Image in front of the digital data.

Each input channel seizes one Instance in the Analog Input Point Object (Class 0x67).

Note

Information for the control/status byte development
Please refer to the corresponding description of the I/O modules for the structure of the control/status bytes. You can find a manual with the relevant I/O module description on the WAGO home page: at: http://www.wago.com .

13.3.3.1 1 Channel Analog Input Modules

750-491, (and all variations)

Table 374: 1 Channel Analog Input Modules

Input Process Image			
Instance	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
n	D1	D0	Measured Value U_D
n+1	D3	D2	Measured Value U_{ref}

The input modules represent 2x2 bytes and seize 2 Instances in Class (0x67).

13.3.3.2 2 Channel Analog Input Modules

750-452, -454, -456, -461, -462, -465, -466, -467, -469, -472, -474, -475, 476, -477, -478, -479, -480, -481, -483, -485, -492, (and all variations),
753-452, -454, -456, -461, -465, -466, -467, -469, -472, -474, -475, 476, -477, 478, -479, -483, -492, (and all variations)

Table 375: 2 Channel Analog Input Modules

Input Process Image			
Instance	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
n	D1	D0	Measured Value Channel 1
n+1	D3	D2	Measured Value Channel 2

The input modules represent 2x2 bytes and seize 2 Instances in Class (0x67).

13.3.3.3 4 Channel Analog Input Modules

750-453, -455, -457, -459, -460, -468, (and all variations),
753-453, -455, -457, -459

Table 376: 4 Channel Analog Input Modules

Input Process Image			
Instance	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
n	D1	D0	Measured Value Channel 1
n+1	D3	D2	Measured Value Channel 2
n+2	D5	D4	Measured Value Channel 3
n+3	D7	D6	Measured Value Channel 4

The input modules represent 4x2 bytes and seize 4 Instances in Class (0x67).

13.3.4 Analog Output Modules

The hardware of an analog output module has 16 bits of measured analog data per channel and 8 bits of control/status. However, the coupler/controller with EtherNet/IP does not have access to the 8 control/status bits. Therefore, the coupler/controller with EtherNet/IP can only access the 16 bits of analog data per channel, which are grouped as words and mapped in Intel format in the Output Process Image.

When digital output modules are also present in the node, the analog output data is always mapped into the Output Process Image in front of the digital data.

Each output channel seizes one Instance in the Analog Output Point Object (Class 0x68).

Information

Information to the structure of the Control/Status byte
For detailed information about the structure of a particular module’s control/status byte, please refer to that module’s manual. Manuals for each module can be found on the Internet under: http://www.wago.com.

13.3.4.1 2 Channel Analog Output Modules

750-550, -552, -554, -556, -560, -562, 563, -585, (and all variations),
753-550, -552, -554, -556

Table 377: 2 Channel Analog Output Modules

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	D1	D0	Output Value Channel 1
n+1	D3	D2	Output Value Channel 2

The output modules represent 2x2 bytes and seize 2 Instances in Class (0x68).

13.3.4.2 4 Channel Analog Output Modules

750-553, -555, -557, -559,
753-553, -555, -557, -559

Table 378: 4 Channel Analog Output Modules

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	D1	D0	Output Value Channel 1
n+1	D3	D2	Output Value Channel 2
n+2	D5	D4	Output Value Channel 3
n+3	D7	D6	Output Value Channel 4

The output modules represent 4x2 bytes and seize 4 Instances in Class (0x68).

13.3.5 Specialty Modules

WAGO has a host of Specialty I/O modules that perform various functions. With individual modules beside the data bytes also the control/status byte is mapped in the process image. The control/status byte is required for the bidirectional data exchange of the module with the higher-ranking control system. The control byte is transmitted from the control system to the module and the status byte from the module to the control system.

This allows, for example, setting of a counter with the control byte or displaying of overshooting or undershooting of the range with the status byte.

The control/status byte always lies in the low byte for the fieldbus coupler/controller with Ethernet/IP.

Information

Information to the structure of the Control/Status byte
For detailed information about the structure of a particular module's control/status byte, please refer to that module's manual. Manuals for each module can be found on the Internet under: http://www.wago.com.

The Specialty Modules represent as analog modules.
For this, the process input data of the Specialty Modules seize one Instance per channel in the Analog Input Point Object (Class 0x67) and the process output data seize one Instance seize one Instance in the Analog Input Point Object (Class 0x67) per channel in the Analog Output Point Object (Class 0x68).

13.3.5.1 Counter Modules

750-404, (and all variations except of /000-005),
753-404, (and variation /000-003)

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/status). The counter value is supplied as 32 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

Table 379: Counter Modules 750-404, (and all variations except of /000-005), 753-404, (and variation /000-003)

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	S	Status byte
	D1	D0	Counter value
	D3	D2	

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C	Control byte
	D1	D0	Counter setting value
	D3	D2	

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-404/000-005

The above Counter Modules have a total of 5 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 1 byte of control/ status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 3 words mapped into each image. Word alignment is applied.

Table 380: Counter Modules 750-404/000-005

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	S	Status byte
	D1	D0	Counter Value of Counter 1
	D3	D2	Counter Value of Counter 2

The specialty modules represent 2x3 bytes input data and seize 2 Instances in Class (0x67).

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C	Control byte
	D1	D0	Counter Setting Value of Counter 1
	D3	D2	Counter Setting Value of Counter 2

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-638,
753-638

The above Counter Modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of counter data and 2 bytes of control/status). The two counter values are supplied as 16 bits. The following tables illustrate the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

Table 381: Counter Modules 750-638, 753-638

Input Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	S0	Status byte von Counter 1
	D1	D0	Counter Value von Counter 1
n+1	-	S1	Status byte von Counter 2
	D3	D2	Counter Value von Counter 2

The specialty modules represent 2x3 bytes input data and seize 2 Instances in Class (0x67).

Output Process Image			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C0	Control byte von Counter 1
	D1	D0	Counter Setting Value von Counter 1
n+1	-	C1	Control byte von Counter 2
	D3	D2	Counter Setting Value von Counter 2

And the specialty modules represent 2x3 bytes output data and seize 2 Instances in Class (0x68).

13.3.5.2 Pulse Width Modules

750-511, (and all variations /xxx-xxx)

The above Pulse Width modules have a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of channel data and 2 bytes of control/ status). The two channel values are supplied as 16 bits. Each channel has its own control/status byte. The following table illustrates the Input and Output Process Image, which has a total of 4 words mapped into each image. Word alignment is applied.

Table 382: Pulse Width Modules 750-511, /xxx-xxx

Input and Output Process			
Instance	Byte Destination		Description
	High Byte	Low Byte	
n	-	C0/S0	Control/Status byte of Channel 1
	D1	D0	Data Value of Channel 1
n+1			Control/Status byte of Channel 2
	D3	D2	Data Value of Channel 2

The specialty modules represent 2x3 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

13.3.5.3 Serial Interface Modules with alternative Data Format

750-650, (and the variations /000-002, -004, -006, -009, -010, -011, -012, -013),
750-651, (and the variations /000-002, -003),
750-653, (and the variations /000-002, -007),
753-650, -653

Note

The process image of the / 003-000-variants depends on the parameterized operating mode!
With the freely parametrizable variations /003 000 of the serial interface modules, the desired operation mode can be set. Dependent on it, the process image of these modules is then the same, as from the appropriate variation.

The above Serial Interface Modules with alternative data format have a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 2 words mapped into each image. Word alignment is applied.

Table 383: Serial Interface Modules with alternative Data Format

Input and Output Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
n	D0	C/S	Data byte	Control/status byte
n+1	D2	D1	Data bytes	

The specialty modules represent 2x2 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

13.3.5.4 Serial Interface Modules with Standard Data Format

750-650/000-001, -014, -015, -016
750-651/000-001
750-653/000-001, -006

The above Serial Interface Modules with Standard Data Format have a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of serial data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have a total of 3 words mapped into each image. Word alignment is applied.

Table 384: Serial Interface Modules with Standard Data Format

Input and Output Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
n	D0	C/S	Data byte	Control/status byte
	D2	D1	Data bytes	
	D4	D3		

The specialty modules represent 1x6 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.5.5 Data Exchange Module

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

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

Table 385: Data Exchange Module

Input and Output Process Image			
Instance	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
n	D1	D0	Data bytes
n+1	D3	D2	

The specialty modules represent 2x2 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

13.3.5.6 SSI Transmitter Interface Modules

750-630, (and all variations)

Note

The process image of the / 003-000-variants depends on the parameterized operating mode!
The operating mode of the configurable /003-000 I/O module versions can be set. Based on the operating mode, the process image of these I/O modules is then the same as that of the respective version.

The above SSI Transmitter Interface modules have a total of 4 bytes of user data in the Input Process Image, which has 2 words mapped into the image.
Word alignment is applied.

Table 386: SSI Transmitter Interface Modules

Input Process Image			
Instance	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
n	D1	D0	Data bytes
n+1	D3	D2	

The specialty modules represent 2x2 bytes input data and seize 2 Instances in Class (0x67).

750-630/000-004, -005, -007

In the input process image, SSI transmitter interface modules with status occupy 5 usable bytes, 4 data bytes, and 1 additional status byte. A total of 3 words are assigned in the process image via word alignment.

Table 387: SSI Transmitter Interface I/O Modules with an Alternative Data Format

Input Process Image				
Instance	Byte Destination High Byte	 High Byte	Description	
n	-	S	not used	Status byte
	D1	D0	Data bytes	
	D3	D2		

The specialty modules represent 1x6 bytes and seize 1 Instance in Class (0x67).

13.3.5.7 Incremental Encoder Interface Modules

750-631/000-004, -010, -011

The above Incremental Encoder Interface modules have 5 bytes of input data and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which have 4 words into each image. Word alignment is applied.

Table 388: Incremental Encoder Interface Modules 750-631/000-004, -010, -011

Input Process Image				
Instance	Byte Destination High Byte	 Low Byte	Description	
n	-	S	not used	Status byte
	D1	D0	Counter word	
	-	-	not used	
	D4	D3	Latch word	

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

Output Process Image			
Instance	Byte Destination High Byte	 Low Byte	Description
n	-	C	Control byte of counter 1
	D1	D0	Counter setting value of counter 1
	-	-	not used
	-	-	not used

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-634

The above Incremental Encoder Interface module has 5 bytes of input data (6 bytes in cycle duration measurement mode) and 3 bytes of output data. The following tables illustrate the Input and Output Process Image, which has 4 words mapped into each image. Word alignment is applied.

Table 389: Incremental Encoder Interface Modules 750-634

Input Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
n	-	S	not used	Status byte
	D1	D0	Counter word	
	-	(D2) *)	not used	(Periodic time)
	D4	D3	Latch word	

*) If cycle duration measurement mode is enabled in the control byte, the cycle duration is given as a 24-bit value that is stored in D2 together with D3/D4.

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

Output Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
n	-	C	not used	Control byte
	D1	D0	Counter setting word	
	-	-	not used	
	-	-		

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

750-637

The above Incremental Encoder Interface Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of encoder data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Table 390: Incremental Encoder Interface Modules 750-637

Input and Output Process Image			
Instance	**Byte Destination**		**Description**
	High Byte	**Low Byte**	
n	-	C0/S0	Control/Status byte of Channel 1
	D1	D0	Data Value of Channel 1
n+1	-	C1/S1	Control/Status byte of Channel 2
	D3	D2	Data Value of Channel 2

The specialty modules represent 2x3 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

750-635,
753-635

The above Digital Pulse Interface module has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following table illustrates the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

Table 391: Incremental Encoder Interface Modules 750-635, 750-635

Input and Output Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
n	D0	C0/S0	Data byte	Control/status byte
	D2	D1	Data bytes	

The specialty modules represent 1x4 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.5.8 DC-Drive Controller

750-636

The DC-Drive Controller maps 6 bytes into both the input and output process image. The data sent and received are stored in up to 4 input and output bytes (D0 ... D3). Two control bytes (C0, C1) and two status bytes (S0/S1) are used to control the I/O module and the drive.

In addition to the position data in the input process image (D0 … D3), it is possible to display extended status information (S2 … S5). Then the three control bytes (C1 … C3) and status bytes (S1 … S3) are used to control the data flow.

Bit 3 of control byte C1 (C1.3) is used to switch between the process data and the extended status bytes in the input process image (Extended Info_ON). Bit 3 of status byte S1 (S1.3) is used to acknowledge the switching process.

Table 392: DC-Drive Controller 750-636

Input Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
n	S1	S0	Status byte S1	Status byte S0
	D1*) / S3**)	D0*) / S2**)	Actual position*) / Extended status byte S3**)	Actual position (LSB) / Extended status byte S2**)
	D3*) / S5**)	D2*) / S4**)	Actual position (MSB) / Extended status byte S3**)	Actual position*) / Extended status byte S4**)

*) ExtendedInfo_ON = ‘0’.
**) ExtendedInfo_ON = ‘1’.

Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	C1	C0	Control byte C1	Control byte C0
	D1	D0	Setpoint position	Setpoint position (LSB)
	D3	D2	Setpoint position (MSB)	Setpoint position

The specialty modules represent 1x6 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.5.9 Steppercontroller

750-670

The Steppercontroller RS422 / 24 V / 20 mA 750-670 provides the fieldbus coupler 12 bytes input and output process image via 1 logical channel. The data to be sent and received are stored in up to 7 output bytes (D0 … D6) and 7 input bytes (D0 … D6), depending on the operating mode.

Output byte D0 and input byte D0 are reserved and have no function assigned.

One I/O module control and status byte (C0, S0) and 3 application control and status bytes (C1 ... C3, S1 ... S3) provide the control of the data flow.

Switching between the two process images is conducted through bit 5 in the control byte (C0 (C0.5). Activation of the mailbox is acknowledged by bit 5 of the status byte S0 (S0.5).

Table 393: Steppercontroller RS 422 / 24 V / 20 mA 750-670

Input Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	reserved	S0	reserved	Status byte S0
	D1	D0	Process data*) / Mailbox**)	
	D3	D2		
	D5	D4		
	S3	D6	Status byte S3	Process data*) / reserved**)
	S1	S2	Status byte S1	Status byte S2

*) Cyclic process image (Mailbox disabled)
**) Mailbox process image (Mailbox activated)

Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	reserved	C0	reserved	Control byte C0
	D1	D0	Process data*) / Mailbox**)	
	D3	D2		
	D5	D4		
	C3	D6	Control byte C3	Process data*) / reserved**)
	C1	C2	Control byte C1	Control byte C2

*) Cyclic process image (Mailbox disabled)
**) Mailbox process image (Mailbox activated)

The specialty modules represent 1x12 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.5.10 RTC Module

750-640

The RTC Module has a total of 6 bytes of user data in both the Input and Output Process Image (4 bytes of module data and 1 byte of control/status and 1 byte ID for command). The following table illustrates the Input and Output Process Image, which have 3 words mapped into each image. Word alignment is applied.

Table 394: RTC Module 750-640

Input and Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	ID	C/S	Command byte	Control/status byte
	D1	D0	Data bytes	
	D3	D2		

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).and seize 1 Instance in Class (0x68).

13.3.5.11 DALI/DSI Master Module

750-641

The DALI/DSI Master module has a total of 6 bytes of user data in both the Input and Output Process Image (5 bytes of module data and 1 byte of control/status). The following tables illustrate the Input and Output Process Image, which have 3 words mapped into each image. Word alignment is applied.

Table 395: DALI/DSI Master module 750-641

Input Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	D0	S	DALI Response	Status byte
	D2	D1	Message 3	DALI Address
	D4	D3	Message 1	Message 2

The specialty modules represent 1x6 bytes input data and seize 1 Instance in Class (0x67).

Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	D0	C	DALI command, DSI dimming value	Control byte
	D2	D1	Parameter 2	DALI Address
	D4	D3	Command extension	Parameter 1

And the specialty modules represent 1x6 bytes output data and seize 1 Instance in Class (0x68).

13.3.5.12 EnOcean Radio Receiver

750-642

The EnOcean radio receiver has a total of 4 bytes of user data in both the Input and Output Process Image (3 bytes of module data and 1 byte of control/status). The following tables illustrate the Input and Output Process Image, which have 2 words mapped into each image. Word alignment is applied.

Table 396: EnOcean Radio Receiver 750-642

Input Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	D0	S	Data byte	Status byte
n+1	D2	D1	Data bytes	

Output Process Image				
Instance	Byte Destination		Description	
	High Byte	Low Byte		
n	-	C	not used	Control byte
n+1	-	-	not used	

The specialty modules represent 2x2 bytes input and output data and seize 2 Instances in Class (0x67) and 2 Instances in Class (0x68).

13.3.5.13 MP Bus Master Module

750-643

The MP Bus Master Module has a total of 8 bytes of user data in both the Input and Output Process Image (6 bytes of module data and 2 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 4 words mapped into each image. Word alignment is applied.

Table 397: MP Bus Master Module 750-643

<table>
<tr><th colspan="5">Input and Output Process Image</th></tr>
<tr><th rowspan="2">Instance</th><th colspan="2">Byte Destination</th><th rowspan="2" colspan="2">Description</th></tr>
<tr><th>High Byte</th><th>Low Byte</th></tr>
<tr><td rowspan="4">n</td><td>C1/S1</td><td>C0/S0</td><td>extended Control/ Status byte</td><td>Control/status byte</td></tr>
<tr><td>D1</td><td>D0</td><td rowspan="3" colspan="2">Data bytes</td></tr>
<tr><td>D3</td><td>D2</td></tr>
<tr><td>D5</td><td>D4</td></tr>
</table>

The specialty modules represent 1x8 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.5.14 *Bluetooth*® RF-Transceiver

750-644

The size of the process image for the *Bluetooth*® module can be adjusted to 12, 24 or 48 bytes.
It consists of a control byte (input) or status byte (output); an empty byte; an overlayable mailbox with a size of 6, 12 or 18 bytes (mode 2); and the *Bluetooth*® process data with a size of 4 to 46 bytes.
Thus, each *Bluetooth*® module uses between 12 and 48 bytes in the process image. The sizes of the input and output process images are always the same.

The first byte contains the control/status byte; the second contains an empty byte.

Process data attach to this directly when the mailbox is hidden. When the mailbox is visible, the first 6, 12 or 18 bytes of process data are overlaid by the mailbox data, depending on their size. Bytes in the area behind the optionally visible mailbox contain basic process data. The internal structure of the *Bluetooth*® process data can be found in the documentation for the *Bluetooth*® 750-644 RF Transceiver.

The mailbox and the process image sizes are set with the startup tool WAGO-I/O-*CHECK*.

Table 398: Bluetooth® RF-Transceiver 750-644

Input and Output Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
n	-	C0/S0	not used	Control/status byte
	D1	D0	Mailbox (0, 3, 6 or 9 words) and Process data (2-23 words)	
	D3	D2		
	D5	D4		
	...	...		
	D45	D44		

The 750-644 constitutes a special module, whose process data (12, 24 or 48 bytes) occupy on instances in classes 0x67 and 0x68.

13.3.5.15 Vibration Velocity/Bearing Condition Monitoring VIB I/O

750-645

The Vibration Velocity/Bearing Condition Monitoring VIB I/O has a total of 12 bytes of user data in both the Input and Output Process Image (8 bytes of module data and 4 bytes of control/status). The following table illustrates the Input and Output Process Image, which have 8 words mapped into each image.
Word alignment is applied.

Table 399: Vibration Velocity/Bearing Condition Monitoring VIB I/O 750-645

Input and Output Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
n	-	C0/S0	not used	Control/status byte (log. Channel 1, Sensor input 1)
	D1	D0	Data bytes (log. Channel 1, Sensor input 1)	
n+1	-	C1/S1	not used	Control/status byte (log. Channel 2, Sensor input 2)
	D3	D2	Data bytes (log. Channel 2, Sensor input 2)	
n+2	-	C2/S2	not used	Control/status byte (log. Channel 3, Sensor input 1)
	D5	D4	Data bytes (log. Channel 3, Sensor input 3)	
n+3	-	C3/S3	not used	Control/status byte (log. Channel 4, Sensor input 2)
	D7	D6	Data bytes (log. Channel 4, Sensor input 2)	

The specialty modules represent 4x3 bytes input and output data and seize 4 Instances in Class (0x67) and 4 Instances in Class (0x68).

13.3.5.16 AS-interface Master Module

750-655

The length of the process image of the AS-interface master module can be set to fixed sizes of 12, 20, 24, 32, 40 or 48 bytes.
It consists of a control or status byte, a mailbox with a size of 0, 6, 10, 12 or 18 bytes and the AS-interface process data, which can range from 0 to 32 bytes.

The AS-interface master module has a total of 6 to maximally 24 words data in both the Input and Output Process Image. Word alignment is applied.

The first Input and output word, which is assigned to an AS-interface master module, contains the status / control byte and one empty byte.
Subsequently the mailbox data are mapped, when the mailbox is permanently superimposed (Mode 1).

In the operating mode with suppressible mailbox (Mode 2), the mailbox and the cyclical process data are mapped next.
The following words contain the remaining process dat.

The mailbox and the process image sizes are set with the startup tool WAGO-I/O-*CHECK*.

Table 400: AS-interface Master module 750-655

Input and Output Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
n	-	C0/S0	not used	Control/status byte
	D1	D0	Mailbox (0, 3, 5, 6 or 9 words)/ Process data (0-16 words)	
	D3	D2		
	D5	D4		
	...	...		
	D45	D44		

The specialty modules represent 1x 12...48 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

13.3.6 System Modules

13.3.6.1 System Modules with Diagnostics

750-610, -611

The modules provide 2 bits of diagnostics in the Input Process Image for monitoring of the internal power supply.

Table 401: System Modules with Diagnostics 750-610, -611

Input Process Image							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
						Diagnostic bit S 2 Fuse	Diagnostic bit S 1 Fuse

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

13.3.6.2 Binary Space Module

750-622

The Binary Space Modules behave alternatively like 2 channel digital input modules or output modules and seize depending upon the selected settings 1, 2, 3 or 4 bits per channel. According to this, 2, 4, 6 or 8 bits are occupied then either in the process input or the process output image.

Table 402: Binary Space Module 750-622 (with behavior like 2 channel digital input)

Input and Output Process Image							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(Data bit DI 8)	(Data bit DI 7)	(Data bit DI 6)	(Data bit DI 5)	(Data bit DI 4)	(Data bit DI 3)	Data bit DI 2	Data bit DI 1

The Binary Space Modules seize 2, 4, 6 or 8 Instances in class (0x65) or in Class (0x66).

14 Application Examples

14.1 Test of MODBUS protocol and fieldbus nodes

You require a MODBUS master to test the function of your fieldbus node. For this purpose, various manufacturers offer a range of PC applications that you can, in part, download from the Internet as free of charge demo versions.

One of the programs which is particularly suitable to test your ETHERNET TCP/IP fieldbus node, is for instance **ModScan** from Win-Tech.

Information

Additional Information
A free of charge demo version from ModScan32 and further utilities from Win-Tech can be found in the Internet under:
http://www.win-tech.com/html/demos.htm

ModScan32 is a Windows application that works as a MODBUS master.

This program allows you to access the data points of your connected ETHERNET TCP/IP fieldbus node and to proceed with the desired changes.

Information

Additional Information
For a description example relating to the software operation, refer to:
http://www.win-tech.com/html/modscan32.htm

14.2 Visualization and Control using SCADA Software

This chapter is intended to give insight into how the WAGO ETHERNET fieldbus coupler/controller can be used for process visualization and control using standard user software.

There is a wide range of process visualization programs, called SCADA Software, from various manufacturers.

Information

Additional Information
For a selection of SCADA products, look under i.e.:
http://www.abpubs.demon.co.uk/scadasites.htm

SCADA is the abbreviation for Supervisory Control and Data Acquisition.

It is a user-orientated tool used as a production information system in the areas of automation technology, process control and production monitoring.

The use of SCADA systems includes the areas of visualization and monitoring, data access, trend recording, event and alarm processing, process analysis and targeted intervention in a process (control).

The WAGO ETHERNET fieldbus node provides the required process input and output values.

Note

SCADA software has to provide a MODBUS device driver and support MODBUS/TCP functions!
When choosing suitable SCADA software, ensure that it provides a MODBUS device driver and supports the MODBUS/TCP functions in the coupler.

Visualization programs with MODBUS device drivers are available from i.e. Wonderware, National Instruments, Think&Do or KEPware Inc., some of which are available on the Internet as demo versions.

The operation of these programs is very specific.
However, a few essential steps are described to illustrate the way an application can be developed using a WAGO ETHERNET fieldbus node and SCADA software in principle:

1. Load the MODBUS ETHERNET driver and select MODBUS ETHERNET

2. Enter the IP address for addressing the fieldbus node

At this point, some programs allow the user to give the node an alias name, i.e. to call the node "Measuring data". The node can then be addressed with this name.

3. Create a graphic object, such as a switch (digital) or a potentiometer (analog)

This object is displayed on the work area.

4. Link the object to the desired data point on the node by entering the following data:

    - Node address (IP address or alias name)
    - The desired MODBUS function codes (register/bit read/write)
    - The MODBUS address of the selected channel

Entry is program specific.

Depending on the user software the MODBUS addressing of a bus module can be represented with up to 5 digits.

Example of the MODBUS Addressing

In the case of SCADA Software Lookout from National Instruments the MODBUS function codes are used with a 6 digit coding, whereby the first digit represents the MODBUS table (0, 1, 3 or 4) and implicit the function code (see following table):

Table 403: MODBUS table and function codes

MODBUS table	MODBUS function code	
0	FC1 or FC15	Reading of input bits or writing of several output bits
1	FC2	Reading of several input bits
3	FC4 or FC 16	Reading of several input registers or writing of several output registers
4	FC3	Reading of several input registers

The following five digits specify the channel number (beginning with 1) of the consecutively numbered digital or analog input and/or output channels.

Examples:

- Reading/writing the first digital input: i.e. 0 0000 1
- Reading/writing the second analog input: i.e. 3 0000 2

Application Example:

Thus, the digital input channel 2 of the above node "Measuring data" can be read out with the input: "Measuring data. 0 0000 2".

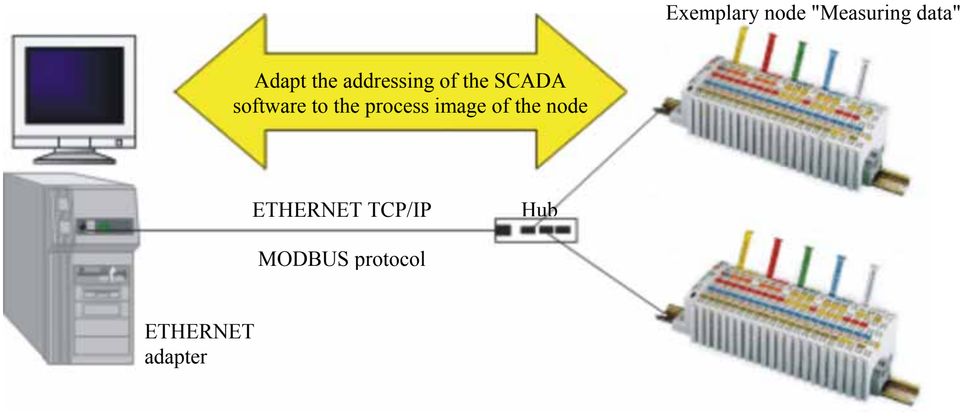


Figure 77: Example SCADA software with MODBUS driver

Information

Additional Information
Please refer to the respective SCADA product manual for a detailed description of the particular software operation.

15 Use in Hazardous Environments

The **WAGO-I/O-SYSTEM 750** (electrical equipment) is designed for use in Zone 2 hazardous areas.

The following sections include both the general identification of components (devices) and the installation regulations to be observed. The individual subsections of the "Installation Regulations" section must be taken into account if the I/O module has the required approval or is subject to the range of application of the ATEX directive.

15.1 Marking Configuration Examples

15.1.1 Marking for Europe according to CENELEC and IEC

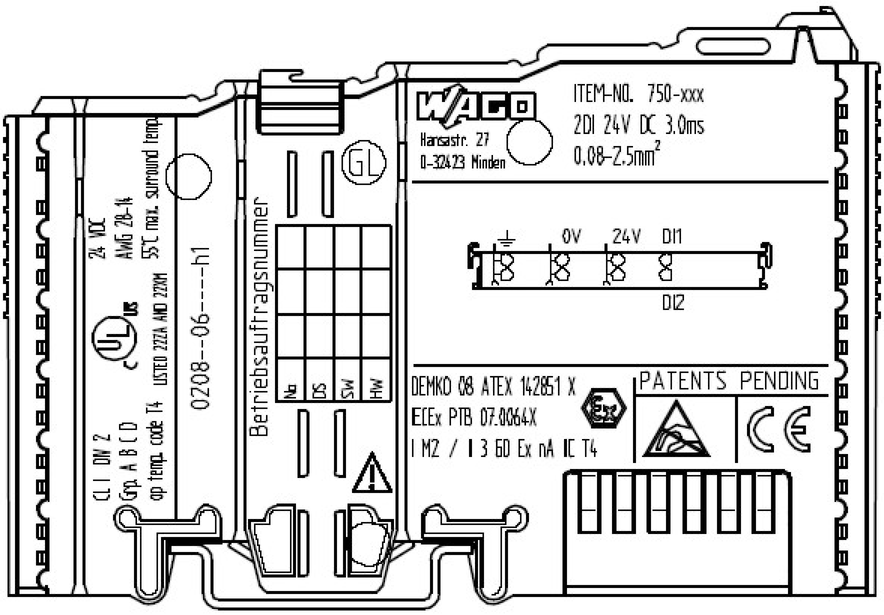


Figure 78: Side marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

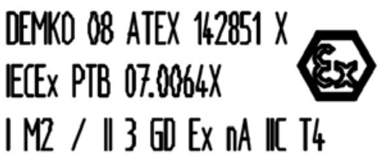


Figure 79: Printing Text detail – Marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

Table 404: Description of marking example for ATEX and IEC Ex approved I/O modules according to CENELEC and IEC

Printing on Text	Description
DEMKO 08 ATEX 142851 X IECEx PTB 07.0064X	Approval body and/or number of the examination certificate
I M2 / II 3 GD	Explosion protection group and Unit category
Ex nA	Type of ignition and extended identification
IIC	Explosion protection group
T4	Temperature class

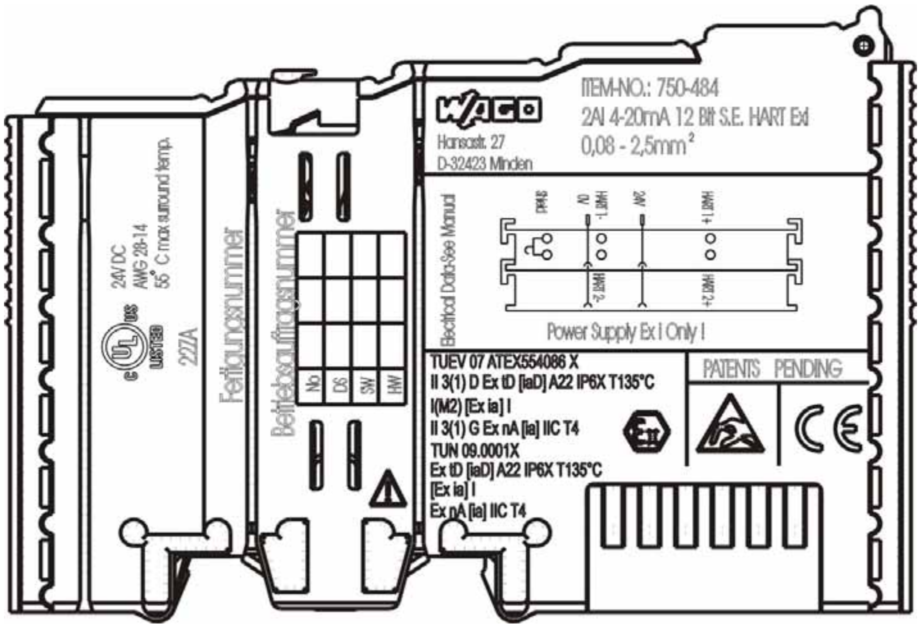


Figure 80: Side marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

TUEV 07 ATEX554086 X
II 3(1) D Ex tD [iaD] A22 IP6X T135°C
I(M2) [Ex ia] I
II 3(1) G Ex nA [ia] IIC T4
TUN 09.0001X
Ex tD [iaD] A22 IP6X T135°C
[Ex ia] I
Ex nA [ia] IIC T4

Figure 81: Text detail – Marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

Table 405: Description of marking example for Ex i and IEC Ex i approved I/O modules according to CENELEC and IEC

Inscription text	Description
TÜV 07 ATEX 554086 X TUN 09.0001X	Approving authority or certificate numbers
Dust	
II	Device group: All except mining
3(1)D	Device category: Zone 22 device (Zone 20 subunit)
Ex	Explosion protection mark
tD	Protection by enclosure
[iaD]	Approved in accordance with "Dust intrinsic safety" standard
A22	Surface temperature determined according to Procedure A, use in Zone 22
IP6X	Dust-tight (totally protected against dust)
T 135°C	Max. surface temp. of the enclosure (no dust bin)
Mining	
I	Device group: Mining
(M2)	Device category: High degree of safety
[Ex ia]	Explosion protection: Mark with category of type of protection intrinsic safety: Even safe when two errors occur
I	Device group: Mining
Gases	
II	Device group: All except mining
3(1)G	Device category: Zone 2 device (Zone 0 subunit)
Ex	Explosion protection mark
nA	Type of protection: Non-sparking operating equipment
[ia]	Category of type of protection intrinsic safety: Even safe when two errors occur
IIC	Explosion Group
T4	Temperature class: Max. surface temperature 135°C

15.1.2 Marking for America according to NEC 500

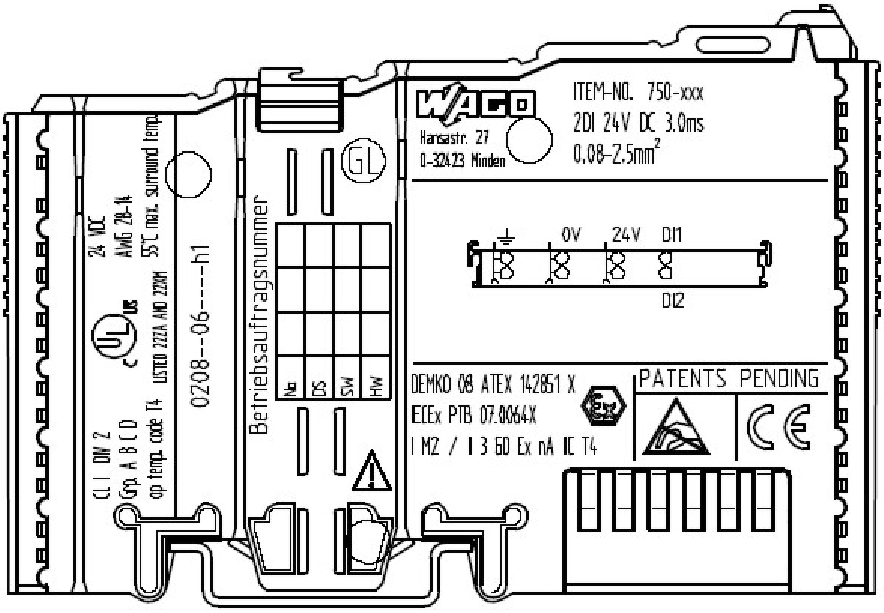


Figure 82: Side marking example for I/O modules according to NEC 500

Figure 83: Text detail – Marking example for I/O modules according to NEC 500

Table 406: Description of marking example for I/O modules according to NEC 500

Printing on Text	Description
CL 1	Explosion protection group (condition of use category)
DIV 2	Area of application (zone)
Grp. ABCD	Explosion group (gas group)
Optemp code T4	Temperature class

15.2 Installation Regulations

In the **Federal Republic of Germany**, various national regulations for the installation in explosive areas must be taken into consideration. The basis for this forms the working reliability regulation, which is the national conversion of the European guideline 99/92/E6. They are complemented by the installation regulation EN 60079-14. The following are excerpts from additional VDE regulations:

Table 407: VDE Installation Regulations in Germany

DIN VDE 0100	Installation in power plants with rated voltages up to 1000 V
DIN VDE 0101	Installation in power plants with rated voltages above 1 kV
DIN VDE 0800	Installation and operation in telecommunication plants including information processing equipment
DIN VDE 0185	lightning protection systems

The **USA** and **Canada** have their own regulations. The following are excerpts from these regulations:

Table 408: Installation Regulations in USA and Canada

NFPA 70	National Electrical Code Art. 500 Hazardous Locations
ANSI/ISA-RP 12.6-1987	Recommended Practice
C22.1	Canadian Electrical Code

NOTICE

Notice the following points
When using the **WAGO-I/O SYSTEM 750** (electrical operation) with Ex approval, the following points are mandatory:

15.2.1 Special Conditions for Safe Operation of the ATEX and IEC Ex (acc. DEMKO 08 ATEX 142851X and IECEx PTB 07.0064)

The fieldbus-independent I/O modules of the WAGO-I/O-SYSTEM 750-.../...-... must be installed in an environment with degree of pollution 2 or better. In the final application, the I/O modules must be mounted in an enclosure with IP 54 degree of protection at a minimum with the following exceptions:

- I/O modules 750-440, 750-609 and 750-611 must be installed in an IP 64 minimum enclosure.
- I/O module 750-540 must be installed in an IP 64 minimum enclosure for 230 V AC applications.
- I/O module 750-440 may be used up to max. 120 V AC.

When used in the presence of combustible dust, all devices and the enclosure shall be fully tested and assessed in compliance with the requirements of IEC 61241-0:2004 and IEC 61241-1:2004.

When used in mining applications the equipment shall be installed in a suitable enclosure according to EN 60079-0:2006 and EN 60079-1:2007.

I/O modules fieldbus plugs or fuses may only be installed, added, removed or replaced when the system and field supply is switched off or the area exhibits no explosive atmosphere.

DIP switches, coding switches and potentiometers that are connected to the I/O module may only be operated if an explosive atmosphere can be ruled out.

I/O module 750-642 may only be used in conjunction with antenna 758-910 with a max. cable length of 2.5 m.

To exceed the rated voltage no more than 40%, the supply connections must have transient protection.

The permissible ambient temperature range is 0 °C to +55 °C.

15.2.2 Special conditions for safe use (ATEX Certificate TÜV 07 ATEX 554086 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the field bus independent I/O modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15, EN 61241-0 and EN 61241-1. For use as group I, electrical apparatus M2, the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.

2. If the interface circuits are operated without the field bus coupler station type 750-3../…-… (DEMKO 08 ATEX 142851 X), measures must be taken outside of the device so that the rating voltage is not being exceeded of more than 40% because of transient disturbances.

3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.

4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “CF-Card”, “USB”, “Fieldbus connection“, “Configuration and programming interface“, “antenna socket“, “D-Sub“ and the “Ethernet interface“. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.

5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.

6. For the type 750-601 the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.

7. The ambient temperature range is: 0°C ≤ T_a ≤ +55°C (for extended details please note certificate).

8. The following warnings shall be placed nearby the unit:

WARNING

Do not remove or replace fuse when energized!
If the module is energized do not remove or replace the fuse.

WARNING

Do not separate when energized!
Do not separate the module when energized!

WARNING

Separate only in a non-hazardous area!
Separate the module only in a non-hazardous area!

15.2.3 Special conditions for safe use (IEC-Ex Certificate TUN 09.0001 X)

1. For use as Dc- or Gc-apparatus (in zone 2 or 22) the fieldbus independent I/O modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15, IEC 61241-0 and IEC 61241-1. For use as group I, electrical apparatus M2, the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64. The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.

2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40% because of transient disturbances.

3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.

4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded. This is although and in particular valid for the interfaces “CF-Card”, “USB”, “Fieldbus connection“, “Configuration and programming interface“, “antenna socket“, “D-Sub“ and the “Ethernet interface“. These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.

5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.

6. For the type 750-601 the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.

7. The ambient temperature range is: $0°C \leq T_a \leq +55°C$ (For extensions please see the certificate).

8. The following warnings shall be placed nearby the unit:

WARNING

Do not remove or replace fuse when energized!
If the module is energized do not remove or replace the fuse.

WARNING

Do not separate when energized!
Do not separate the module when energized!

WARNING

Separate only in a non-hazardous area!
Separate the module only in a non-hazardous area!

15.2.4 ANSI/ISA 12.12.01

This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only.

This equipment is to be fitted within tool-secured enclosures only.

⚠ WARNING

Explosion hazard!
Explosion hazard - substitution of components may impair suitability for Class I, Div. 2.

⚠ WARNING

Disconnect device when power is off and only in a non-hazardous area!
Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous near each operator accessible connector and fuse holder." When a fuse is provided, the following information shall be provided: "A switch suitable for the location where the equipment is installed shall be provided to remove the power from the fuse."

For devices with Ethernet connectors:
"Only for use in LAN, not for connection to telecommunication circuits".

⚠ WARNING

Use only with antenna module 758-910!
Use Module 750-642 only with antenna module 758-910.

For Couplers/Controllers and Economy bus modules only: "The configuration Interface Service connector is for temporary connection only. Do not connect or disconnect unless the area is known to be nonhazardous. Connection or disconnection in an explosive atmosphere could result in an explosion.

⚠ WARNING

Devices containing fuses must not be fitted into circuits subject to over loads!
Devices containing fuses must not be fitted into circuits subject to over loads, e.g. motor circuits!

WARNING

Do not connect or disconnect SD-Card unless the area known to be free of ignitable concentrations of flammable gases or vapors!
Do not connect or disconnect SD-Card while circuit is live unless the area is known to be free of ignitable concentrations of flammable gases or vapors.

Information

Additional Information
Proof of certification is available on request. Also take note of the information given on the module technical information sheet. The Instruction Manual, containing these special conditions for safe use, must be readily available to the user.

16 Appendix

16.1 MIB II Groups

16.1.1 System Group

The system group contains general information about the coupler/controller.

Table 409: MIB II – System group

Identifier	Entry	Access	Description
1.3.6.1.2.1.1.1	sysDescr	R	This entry contains the device identification. The object has a fixed code (e.g., "WAGO 750-841")
1.3.6.1.2.1.1.2	sysObjectID	R	This entry contains the manufacturer's authorization identification.
1.3.6.1.2.1.1.3	sysUpTime	R	This entry contains the time (in hundredths of a second) since the management unit has been last reset.
1.3.6.1.2.1.1.4	sysContakt	R/W	This entry contains the identification and contact information for the system contact person.
1.3.6.1.2.1.1.5	sysName	R/W	This entry contains the administration-assigned device name.
1.3.6.1.2.1.1.6	sysLocation	R/W	This entry contains the node’s physical location.
1.3.6.1.2.1.1.7	sysServices	R	This entry designates the quantity of services that this coupler/controller contains.

16.1.2 Interface Group

The interface group contains information and statistics about the device interface.

A device interface describes the Ethernet interface of a coupler/controller and provides status information on the physical Ethernet ports as well as on the internal loopback interface.

Table 410: MIB II – Interface Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.2.1	ifNumber	R	Number of network interfaces in this system
1.3.6.1.2.1.2.2	ifTable	-	List of network interfaces
1.3.6.1.2.1.2.2.1	ifEntry	-	Network interface entry
1.3.6.1.2.1.2.2.1.1	ifIndex	R	This entry contains a unique value for each interface
1.3.6.1.2.1.2.2.1.2	ifDescr	R	This entry contains the name of the manufacturer, the product name, and the version of the hardware interface: e.g., "WAGO Kontakttechnik GmbH 750-841: Rev 1.0"
1.3.6.1.2.1.2.2.1.3	ifType	R	This entry describes the type of interface. ETHERNET CSMA/CD = 6 Software Loopback = 24
1.3.6.1.2.1.2.2.1.4	ifMtu	R	This entry specifies the largest transfer unit; i.e., the maximum telegram length that can be transferred via this interface.
1.3.6.1.2.1.2.2.1.5	ifSpeed	R	This entry indicates the interface speed in bits per second.
1.3.6.1.2.1.2.2.1.6	ifPhysAddress	R	This entry indicates the physical address of the interface. For example, for Ethernet, this entry contains a MAC ID.
1.3.6.1.2.1.2.2.1.7	ifAdmin-Status	R/W	This entry specifies the desired state of the interfaces. Possible values are: up(1): Ready for operation for transmission and reception down(2): Interface is switched off testing(3): Interface is in test mode
1.3.6.1.2.1.2.2.1.8.	ifOperStatus	R	This entry indicates the current operational state of the interface.
1.3.6.1.2.1.2.2.1.9.	ifLastChange	R	This entry indicates the value of the sysUpTime when the state was last changed.
1.3.6.1.2.1.2.2.1.10	ifInOctets	R	This entry gives the total number of bytes received via interface.
1.3.6.1.2.1.2.2.1.11	ifInUcastPkts	R	This entry indicates the number of received unicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.12	ifInNUcastPkts	R	This entry indicates the number of received broad and multicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.13	ifInDiscards	R	This entry indicates the number of packets that were discarded even though no errors had been detected.
1.3.6.1.2.1.2.2.1.14	ifInErrors	R	This entry indicates the number of received packets that contained errors preventing them from being deliverable to a higher layer.

Table 410: MIB II – Interface Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.2.2.1.15	IfInUnknown-Protos	R	This entry indicates the number of received packets sent to an unknown or unsupported port number.
1.3.6.1.2.1.2.2.1.16	ifOutOctets	R	This entry gives the total number of bytes sent via interface.
1.3.6.1.2.1.2.2.1.17	ifOutUcastPkts	R	This entry contains the number of outgoing unicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.18	ifOutNUcastPkts	R	This entry indicates the number of outgoing broad and multicast packets delivered to a higher layer.
1.3.6.1.2.1.2.2.1.19	ifOutDiscards	R	This entry indicates the number of packets that were discarded even though no errors had been detected.
1.3.6.1.2.1.2.2.1.20	ifOutErrors	R	This entry indicates the number of packets that could not be transmitted because of errors.

16.1.3 IP Group

The IP group contains information about IP communication.

Table 411: MIB II – IP Group

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

16.1.4 IpRoute Table Group

The IP route table contains information about the routing table in the coupler/controller.

Table 412: MIB II – IpRoute Table Group

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

16.1.5 ICMP Group

Table 413: MIB II – ICMP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.5.1	icmpInMsgs	R	Number of received ICMP messages
1.3.6.1.2.1.5.2	icmpInErrors	R	Number of received ICMP errors containing ICMP-specific errors
1.3.6.1.2.1.5.3	icmpInDestUnreachs	R	Number of received ICMP destination unreachable messages
1.3.6.1.2.1.5.4	icmpInTimeExcds	R	Number of received ICMP time exceeded messages
1.3.6.1.2.1.5.5	icmpInParmProbs	R	Number of received ICMP parameter problem messages
1.3.6.1.2.1.5.6	icmpInSrcQuenchs	R	Number of received ICMP source quench messages
1.3.6.1.2.1.5.7	icmpInRedirects	R	Number of received ICMP redirect messages
1.3.6.1.2.1.5.8	icmpInEchos	R	Number of received ICMP echo request messages (Ping)
1.3.6.1.2.1.5.9	icmpInEchoReps	R	Number of received ICMP echo reply messages (Ping)
1.3.6.1.2.1.5.10	icmpInTimestamps	R	Number of received ICMP timestamp request messages
1.3.6.1.2.1.5.11	icmpInTimestampReps	R	Number of received ICMP timestamp reply messages
1.3.6.1.2.1.5.12	icmpInAddrMasks	R	Number of received ICMP address mask request messages
1.3.6.1.2.1.5.13	icmpInAddrMaskReps	R	Number of received ICMP address mask reply messages
1.3.6.1.2.1.5.14	icmpOutMsgs	R	Number of sent ICMP messages
1.3.6.1.2.1.5.15	icmpOutErrors	R	Number of sent ICMP messages that could not be sent due to errors
1.3.6.1.2.1.5.16	icmpOutDestUnreachs	R	Number of sent ICMP destination unreachable messages
1.3.6.1.2.1.5.17	icmpOutTimeExcds	R	Number of sent ICMP time exceeded messages
1.3.6.1.2.1.5.18	icmpOutParmProbs	R	Number of sent ICMP parameter problem messages
1.3.6.1.2.1.5.19	icmpOutSrcQuenchs	R	Number of sent ICMP source quench messages
1.3.6.1.2.1.5.20	icmpOutRedirects	R	Number of sent ICMP redirection messages
1.3.6.1.2.1.5.21	icmpOutEchos	R	Number of sent ICMP echo request messages
1.3.6.1.2.1.5.22	icmpOutEchoReps	R	Number of sent ICMP echo reply messages
1.3.6.1.2.1.5.23	icmpOutTimestamps	R	Number of sent ICMP timestamp request messages
1.3.6.1.2.1.5.24	icmpOutTimestampReps	R	Number of sent ICMP timestamp reply messages
1.3.6.1.2.1.5.25	icmpOutAddrMasks	R	Number of sent ICMP address mask request messages
1.3.6.1.2.1.5.26	icmpOutAddrMaskReps	R	Number of sent ICMP address mask reply messages

16.1.6 TCP Group

Table 414: MIB II – TCP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.6.1	tcpRtoAlgorithm	R	Retransmission time: 1 = other, 2 = constant, 3 = RSRE, 4 = VANJ
1.3.6.1.2.1.6.2	tcpRtoMin	R	Minimum value for the retransmission timer
1.3.6.1.2.1.6.3	tcpRtoMax	R	Maximum value for the retransmission timer
1.3.6.1.2.1.6.4	tcpMaxConn	R	Number of maximum TCP connections that can exist simultaneously
1.3.6.1.2.1.6.5	tcpActiveOpens	R	Number of existing active TCP connections
1.3.6.1.2.1.6.6	tcpPassiveOpens	R	Number of existing passive TCP connections
1.3.6.1.2.1.6.7	tcpAttemptFails	R	Number of failed connection attempts
1.3.6.1.2.1.6.8	tcpEstabResets	R	Number of connection resets
1.3.6.1.2.1.6.9	tcpCurrEstab	R	The number of TCP connections for which the current state is either Established or Close-Wait
1.3.6.1.2.1.6.10	tcpInSegs	R	Number of received TCP frames including the error frames
1.3.6.1.2.1.6.11	tcpOutSegs	R	Number of correctly sent TCP frames with data
1.3.6.1.2.1.6.12	tcpRetransSegs	R	Number of sent TCP frames retransmitted because of errors
1.3.6.1.2.1.6.13	tcpConnTable	-	For each existing connection, a table entry is created
1.3.6.1.2.1.6.13.1	tcpConnEntry	-	Table entry for connection
1.3.6.1.2.1.6.13.1.1	tcpConnState	R	This entry indicates the status of the TCP connection
1.3.6.1.2.1.6.13.1.2	tcpConnLocalAddress	R	The entry contains the IP address for the connection. For a server, this entry is constant 0.0.0.0
1.3.6.1.2.1.6.13.1.3	tcpConnLocalPort	R	The entry indicates the port number of the TCP connection.
1.3.6.1.2.1.6.13.1.4	tcpConnRemAddress	R	The entry contains the remote IP address of the TCP connection.
1.3.6.1.2.1.6.13.1.5	tcpConnRemPort	R	The entry contains the remote port of the TCP connection.
1.3.6.1.2.1.6.14	tcpInErrs	R	Number of received incorrect TCP frames
1.3.6.1.2.1.6.15	tcpOutRsts	R	Number of sent TCP frames with set RST flag

16.1.7 UDP Group

Table 415: MIB II – UDP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.7.1	udpInDatagrams	R	Number of received UDP frames that could be passed on to the appropriate applications
1.3.6.1.2.1.7.2	udpNoPorts	R	Number of received UDP frames that could not be passed on to the appropriate applications (port unreachable)
1.3.6.1.2.1.7.3	udpInErrors	R	Number of received UDP frames that could not be passed on to the appropriate applications for other reasons.
1.3.6.1.2.1.7.4	udpOutDatagrams	R	Number of sent UDP frames
1.3.6.1.2.1.7.5	udpTable	-	A table entry is created for each application that received UDP frames
1.3.6.1.2.1.7.5.1	udpEntry	-	Table entry for an application that received an UDP frame
1.3.6.1.2.1.7.5.1.1	udpLocalAddress	R	IP address of the local UDP server
1.3.6.1.2.1.7.5.1.2	udpLocalPort	R	Port number of the local UDP server

16.1.8 SNMP Group

Table 416: MIB II – SNMP Group

Identifier	Entry	Access	Description
1.3.6.1.2.1.11.1	snmpInPkts	R	Number of received SNMP frames
1.3.6.1.2.1.11.2	snmpOutPkts	R	Number of sent SNMP frames
1.3.6.1.2.1.11.3	snmpInBadVersions	R	Number of received SNMP frames with an invalid version number
1.3.6.1.2.1.11.4	snmpInBadCommunity-Names	R	Number of received SNMP frames with an invalid community
1.3.6.1.2.1.11.5	snmpInBadCommunity Uses	R	Number of received SNMP frames whose community did not have sufficient authorization for the actions that it tried to execute
1.3.6.1.2.1.11.6	snmpInASNParseErrs	R	Number of received SNMP frames with an incorrect structure
1.3.6.1.2.1.11.8	snmpInTooBigs	R	Number of received SNMP frames that acknowledged the result too Big
1.3.6.1.2.1.11.9	snmpInNoSuchNames	R	Number of received SNMP frames that acknowledged the result noSuchName
1.3.6.1.2.1.11.10	snmpInBadValues	R	Number of received SNMP frames that acknowledged the result bad value
1.3.6.1.2.1.11.11	snmpInReadOnlys	R	Number of received SNMP frames that acknowledged the result readOnly
1.3.6.1.2.1.11.12	snmpInGenErrs	R	Number of received SNMP frames that acknowledged the result genError
1.3.6.1.2.1.11.13	snmpInTotalReqVars	R	Number of received SNMP frames with valid GET or GET-NEXT requests
1.3.6.1.2.1.11.14	snmpInTotalSetVars	R	Number of received SNMP frames with valid SET requests
1.3.6.1.2.1.11.15	snmpInGetRequests	R	Number of GET requests received and processed
1.3.6.1.2.1.11.16	snmpInGetNexts	R	Number of GET-NEXT requests received and processed
1.3.6.1.2.1.11.17	snmpInSetRequests	R	Number of SET requests received and processed
1.3.6.1.2.1.11.18	snmpInGetResponses	R	Number of received GET responses
1.3.6.1.2.1.11.19	snmpInTraps	R	Number of received traps
1.3.6.1.2.1.11.20	snmpOutTooBigs	R	Number of sent SNMP frames that contained the result too Big
1.3.6.1.2.1.11.21	snmpOutNoSuchNames	R	Number of sent SNMP frames that contained the result noSuchName
1.3.6.1.2.1.11.22	snmpOutBadValues	R	Number of sent SNMP frames that contained the result bad value
1.3.6.1.2.1.11.24	SnmpOutGenErrs	R	Number of sent SNMP frames that contained the result genErrs
1.3.6.1.2.1.11.25	snmpOutGetRequests	R	Number of GET requests sent
1.3.6.1.2.1.11.26	SnmpOutGetNexts	R	Number of GET NEXT requests sent
1.3.6.1.2.1.11.27	snmpOutSetRequests	R	Number of SET requests sent
1.3.6.1.2.1.11.28	snmpOutGetResponses	R	Number of GET responses sent
1.3.6.1.2.1.11.29	snmpOutTraps	R	Number of traps sent
1.3.6.1.2.1.11.30	snmpEnableAuthenTraps	R/W	Authentification failure traps(1 = on, 2 = off)

16.2 WAGO MIB Groups

16.2.1 Company Group

The company group contains general information about the company WAGO Kontakttechnik GmbH & Co. KG.

Table 417: WAGO MIB – Company Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.1.1	wagoName	R	Company's registered name Default value: "WAGO Kontakttechnik GmbH & Co. KG"
1.3.6.1.4.1.13576.1.2	wagoDescrition	R	Description of company Default value: "WAGO Kontakttechnik GmbH & Co. KG, Hansastr. 27, D-32423 Minden"
1.3.6.1.4.1.13576.1.3	wagoURL	R	URL for company web site Default value: "www.wago.com"

16.2.2 Product Group

The product group contains information about the controller.

Table 418: WAGO MIB – Product Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.1	wioArticleName	R	Name of article Default value: "750-8xx/000-000"
1.3.6.1.4.1.13576.10.1.2	wioArticleDescription	R	Description of article Default value: "WAGO Ethernet (10/100MBit) FBC"
1.3.6.1.4.1.13576.10.1.3	wioSerialNumber	R	Serial number of article Default value: "SNxxxxxxxx-Txxxxxx-mac|0030DExxxxxx"
1.3.6.1.4.1.13576.10.1.4	wioMacAddress	R	MAC address of article Default value: "0030DExxxxxx"
1.3.6.1.4.1.13576.10.1.5	wioURLDatasheet	R	URL to datasheet of article Default value: "http://www.wago.com/ wagoweb/documentation/navigate/nm0dc__e.htm#ethernet"
1.3.6.1.4.1.13576.10.1.6	wioURLManual	R	URL to manual of article Default value: "http://www.wago.com/ wagoweb/documentation/navigate/nm0dc__e.htm#ethernet"

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.7	wioDeviceClass	R	Device class 10 = controller 20 = coupler 30 = switch 40 = display 50 = sensor 60 = actuator
1.3.6.1.4.1.13576.10.1.8	wioDeviceGroup	R	Device group 10 = Serie 750 20 = Serie 758 30 = Serie 767 40 = Serie 762 PERSPECTO

16.2.3 Versions Group

The version group contains information about the hardware/software versions used in the controller.

Table 419: WAGO MIB – Versions Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.10.1	wioFirmwareIndex	R	Index of firmware version
1.3.6.1.4.1.13576.10.1.10.2	wioHardwareIndex	R	Index of hardware version
1.3.6.1.4.1.13576.10.1.10.3	wioFwlIndex	R	Index of software version from firmware loader
1.3.6.1.4.1.13576.10.1.10.4	wioFirmwareVersion	R	Complete firmware string

16.2.4 Real-Time Clock Group

The real-time clock group contains information about the system’s real-time clock.

Table 420: WAGO MIB – Real Time Clock Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.11.1	wioRtcDateTime	R/W	Date/time of coupler in UTC as string. For writing date/time use the following string time 11:22:33 date 13-1-2007 Default value: "time xx:xx:xx date xx-xx-xxxx (UTC)"
1.3.6.1.4.1.13576.10.1.11.2	wioRtcTime	R/W	Date/time of coupler in UTC as integer in seconds from 1970-01-01 Default value: “0”
1.3.6.1.4.1.13576.10.1.11.3	wioTimezone	R/W	"Actual time zone of article in hours (-12 - +12) Default value: “0”
1.3.6.1.4.1.13576.10.1.11.4	wioRtcHourMode	R	Hour mode 0 = 12h mode 1 = 24h mode Default value: “0”
1.3.6.1.4.1.13576.10.1.11.5	wioRtcBatteryStatus	R	RTC battery status: 0 = ok 1 = battery empty Default value: “1”
1.3.6.1.4.1.13576.10.1.11.6	wioRtcDayLightSaving	R/W	Time offset of 1 hour: 0 = not offset 1 = offset 1 hour (DayLightSaving) Default value: “0”

16.2.5 Ethernet Group

The Ethernet group contains the settings for the controller on the Ethernet.

Table 421: WAGO MIB – Ethernet Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.12.1	wioEthernetMode	R/W	IP configuration of Ethernet connection: 0 = fix Ip address 1 = dynamic IP address over Bootp 2 = dynamic IP address over DHCP Default value: “1”
1.3.6.1.4.1.13576.10.1.12.2	wioIp	R/W	Actual IP address of coupler
1.3.6.1.4.1.13576.10.1.12.3	wioSubnetMask	R/W	Actual subnet mask of coupler
1.3.6.1.4.1.13576.10.1.12.4	wioGateway	R/W	Actual gateway IP of coupler
1.3.6.1.4.1.13576.10.1.12.5	wioHostname	R/W	Actual host name of coupler
1.3.6.1.4.1.13576.10.1.12.6	wioDomainName	R/W	Actual domain name of coupler
1.3.6.1.4.1.13576.10.1.12.7	wioDnsServer1	R/W	IP address of first DNS server
1.3.6.1.4.1.13576.10.1.12.8	wioDnsServer2	R/W	IP address of second DNS server

16.2.6 Actual Error Group

The actual error group contains information about the last system status/error status.

Table 422: WAGO MIB – Actual Error Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.20.1	wioErrorGroup	R	Error group of last error
1.3.6.1.4.1.13576.10.1.20.2	wioErrorCode	R	Error code of last error
1.3.6.1.4.1.13576.10.1.20.3	wioErrorArgument	R	Error argument of last error
1.3.6.1.4.1.13576.10.1.20.4	wioErrorDescription	R	Error description string

16.2.7 PLC Project Group

The PLC project group contains information about the controller's PLC program.

Table 423: WAGO MIB – PLC Project Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.30.1	wioProjectId	R	ID of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.2	wioProjectDate	R	Date of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.3	wioProjectName	R	Name of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.4	wioProjectTitle	R	Title of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.5	wioProjectVersion	R	Version of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.6	wioProjectAuthor	R	Author of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.7	wioProjectDescription	R	Description of CoDeSys project
1.3.6.1.4.1.13576.10.1.30.8	wioNumberOfIecTasks	R	Number of IEC tasks in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9	wioIecTaskTable	-	
1.3.6.1.4.1.13576.10.1.30.9.1	wioIecTaskEntry	-	
1.3.6.1.4.1.13576.10.1.30.9.1.1	wioIecTaskId	R	ID of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.2	wioIecTaskName	R	Name of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.3	wioIecTaskStatus	R	Status of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.4	wioIecTaskMode	R	Mode of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.5	wioIecTaskPriority	R	Priority of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.6	wioIecTaskInterval	R	Interval of cyclic IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.7	wioIecTaskEvent	R	Event for IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.8	wioIecTaskCycleCount	R	Count of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.9	wioIecTaskCycleTime	R	Last cycle time of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.10	wioIecTaskCycleTime-Min	R	Minimal cycle time of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.11	wioIecTaskCycleTime-Max	R	Maximal cycle time of IEC task in the CoDeSys project
1.3.6.1.4.1.13576.10.1.30.9.1.12	wioIecTaskCycleTime-Avg	R	Average cycle time of IE task in the CoDeSys project

16.2.8 Http Group

The Http group contains information and settings for the controller's Web server.

Table 424: WAGO MIB – Http Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.1.1	wioHttpEnable	R/W	Enable/disable the port of the webserver: 0 = port of webserver disable 1 = port of webserver enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.1.2	wioHttpAuthen-ticationEnable	R/W	Enable/disable the authentication on the websides: 0 = authentication disabled 1 = authentication enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.1.3	wioHttpPort	R/W	Port of the http web server Default value: { 80 }

16.2.9 Ftp Group

The Ftp group contains information and settings for the controller's Ftp server.

Table 425: WAGO MIB – Ftp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.2.1	wioFtpEnable	R/W	Enable/disable the port of the ftp server 0 = port of ftp server disable 1 = port of ftp server enable Default value: { 1 }

16.2.10 Sntp Group

The Sntp group contains information and settings for the controller's Sntp server.

Table 426: WAGO MIB – Sntp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.3.1	wioSntpEnable	R/W	Enable/disable the port of the SNTP server 0 = port of SNTP server disable 1 = port of SNTP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.3.2	wioSntpServer-Address	R/W	IP address of SNTP server Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.3.3	wioSntpClient-Intervall	R/W	Interval to pool SNTP manager Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.3.4	wioSntpClient-Timeout	R/W	Timeout to corrupt SNTP answer Default value: { 2000 }
1.3.6.1.4.1.13576.10.1.40.3.5	wioSntpClient-DayLightSaving	R/W	Time offset of 1 hour: 0 = not offset 1 = offset 1 hour (DayLightSaving) Default value: { 0 }

16.2.11 Snmp Group

The Snmp group contains information and settings for the controller's SNMP agent.

Table 427: WAGO MIB – Snmp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.1	wioSnmpEnable	R/W	Enable/disable the port of the SNMP server 0 = port of SNMP server disable 1 = port of SNMP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.2.1	wioSnmp1-ProtocolEnable	R/W	Enable/disable first SNMPv1/v2c agent Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.2.2	wioSnmp1-ManagerIp	R/W	IP address of first SNMP server Default value: { 'C0A80101'h }
1.3.6.1.4.1.13576.10.1.40.4.2.3	wioSnmp1-Community	R/W	Community identification string for SNMPv1/v2c Default value: { "public" }
1.3.6.1.4.1.13576.10.1.40.4.2.4	wioSnmp1Trap-V1enable	R/W	Enable/disable SNMPv1 traps to first SNMP server Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.2.5	wioSnmp1Trap-V2enable	R/W	Enable/disable SNMPv2c traps to first SNMP server Default value: { 0 }

Table 427: WAGO MIB – Snmp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.2.6	wioSnmp2-ProtocolEnable	R/W	Enable/disable first SNMPv1/v2c agent Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.2.7	wioSnmp2-ManagerIp	R/W	IP address of second SNMP server Default value: { '00000000'h }
1.3.6.1.4.1.13576.10.1.40.4.2.8	wioSnmp2-Community	R/W	Community identification string for SNMPv1/v2c Default value: { "public" }
1.3.6.1.4.1.13576.10.1.40.4.2.9	wioSnmp2Trap-V1enable	R/W	Enable/disable SNMPv1 traps to first SNMP server Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.2.10	wioSnmp2Trap-V2enable	R/W	Enable/disable SNMPv2c traps to first SNMP server Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.3.1	wioSnmp1User-Enable	R/W	Enable/disable first SNMPv3 user Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.2	wioSnmp1-Authentication-Typ	R/W	Athentication typ for first SNMPv3 user: 0 = no Authentication 1 = MD5 Authentication 2 = SHA1 Authentication Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.3	wioSnmp1-Authentication-Name	R/W	Authentication name for first SNMPv3 user Default value: { "SecurityName" }
1.3.6.1.4.1.13576.10.1.40.4.3.4	wioSnmp1-Authentication-Key	R/W	Authentication key for first SNMPv3 user Default value: {"AuthenticationKey "}
1.3.6.1.4.1.13576.10.1.40.4.3.5	wioSnmp1-PrivacyEnable	R/W	Disable/enable data encryption for first SNMPv3 user:: 0 = no Encryption 1 = DES Encryption Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.6	wioSnmp1-PrivacyKey	R/W	Privacy key for SNMPv3 for first SNMPv3 user Default value: { "PrivacyKey" }
1.3.6.1.4.1.13576.10.1.40.4.3.7	wioSnmp1-Notification-Enable	R/W	Enable/disable notification (SNMPv3 traps) with SNMPv3 user Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.8	wioSnmp1-Notification-ReceiverIP	R/W	Receiver IP address for notification (SNMPv3 traps) with SNMPv3 user Default value: { 'C0A80101'h }
1.3.6.1.4.1.13576.10.1.40.4.3.9	wioSnmp2User-Enable	R/W	Enable/disable second SNMPv3 user Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.3.10	wioSnmp2-Authentication-Typ	R/W	Authentication typ for second SNMPv3 user: 0 = no authentication 1 = MD5 authentication 2 = SHA1 authentication Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.11	wioSnmp2-Authentication-Name	R/W	Authentication name for second SNMPv3 user Default value: { "SecurityName" }

Table 427: WAGO MIB – Snmp Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.3.12	wioSnmp2-Authentication-Key	R/W	Authentication key for second SNMPv3 user Default value: {"AuthenticationKey"}
1.3.6.1.4.1.13576.10.1.40.4.3.13	wioSnmp2-PrivacyEnable	R/W	Privacy key for SNMPv3 for second SNMPv3 user Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.4.3.14	wioSnmp2-PrivacyKey	R/W	Privacy key for SNMPv3 for second SNMPv3 user Default value: { "PrivacyKey" }
1.3.6.1.4.1.13576.10.1.40.4.3.15	wioSnmp2-Notification-Enable	R/W	Enable/disable notification (SNMPv3 traps) with SNMPv3 user Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.4.3.16	wioSnmp2-Notification-ReceiverIP	R/W	Receiver IP address for notification (SNMPv3 traps) with SNMPv3 user Default value: { '00000000'h }

16.2.12 Snmp Trap String Group

The Snmp trap string group contains strings that are attached to the manufacturer-specific traps.

Table 428: WAGO MIB – Snmp Trap String Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.4.1	wioTrapKbus-Error	R/W	String for 1st SNMP trap Default value: { "Kbus Error" }
1.3.6.1.4.1.13576.10.1.40.4.4.2	wioTrapPlcStart	R/W	String for 2nd SNMP trap Default value: { "Plc Start" }
1.3.6.1.4.1.13576.10.1.40.4.4.3	wioTrapPlcStop	R/W	String for 3rd SNMP trap Default value: { "Plc Stop" }
1.3.6.1.4.1.13576.10.1.40.4.4.4	wioTrapPlc-Reset	R/W	String for 4th SNMP trap Default value: { "Plc Reset" }
1.3.6.1.4.1.13576.10.1.40.4.4.5	wioTrapPlcSoftwareWatchdog	R/W	String for 5th SNMP trap Default value: { "Plc Software Watchdog" }
1.3.6.1.4.1.13576.10.1.40.4.4.6	wioTrapPlc-DivideByZero	R/W	String for 6th SNMP trap Default value: {"Plc Divide By Zero"}
1.3.6.1.4.1.13576.10.1.40.4.4.7	wioTrapPlc-OnlineChange	R/W	String for 7th SNMP trap Default value: {"Plc Online Change"}
1.3.6.1.4.1.13576.10.1.40.4.4.8	wioTrapPlc-Download	R/W	String for 8th SNMP trap Default value: { "Plc Download Programm" }
1.3.6.1.4.1.13576.10.1.40.4.4.9	wioTrapPlc-Login	R/W	String for 9th SNMP trap Default value: { "Plc Login" }
1.3.6.1.4.1.13576.10.1.40.4.4.10	wioTrapPlc-Logout	R/W	String for 10th SNMP trap Default value: {"Plc Logout"}

16.2.13 Snmp User Trap String Group

The Snmp user trap string group contains strings that can be attached to user-specific traps. These strings can be changed via SNMP or Wago_SNMP.lib in CoDeSys.

Table 429: WAGO MIB – Snmp User Trap String Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.4.5.1	wioUserTrapMsg1	R/W	String for 1st SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.2	wioUserTrapMsg2	R/W	String for 2nd SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.3	wioUserTrapMsg3	R/W	String for 3rd SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.4	wioUserTrapMsg4	R/W	String for 4th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.5	wioUserTrapMsg5	R/W	String for 5th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.6	wioUserTrapMsg6	R/W	String for 6th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.7	wioUserTrapMsg7	R/W	String for 7th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.8	wioUserTrapMsg8	R/W	String for 8th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.9	wioUserTrapMsg9	R/W	String for 9th SNMP trap
1.3.6.1.4.1.13576.10.1.40.4.5.10	wioUserTrapMsg10	R/W	String for 10th SNMP trap

16.2.14 Plc Connection Group

Activate or deactivate the connection to CoDeSys with the Plc connection group.

Table 430: WAGO MIB – Plc Connection Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.5.1	wioCoDeSysEnable	R/W	Enable/disable the port of the CoDeSys server 0 = port of CoDeSys server disable 1 = port of CoDeSys server enable Default value: { 1 }

16.2.15 Modbus Group

The Modbus group contains information and settings about the controller's modbus server.

Table 431: WAGO MIB – Modbus Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.6.1	wioModbusTcp-Enable	R/W	Enable/disable the port of the Modbus TCP server 0 = port of Modbus TCP server disable 1 = port of Modbus TCP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.6.2	wioModbusUdb-Enable	R/W	Enable/disable the port of the Modbus UDP server 0 = port of Modbus UDP server disable 1 = port of Modbus UDP server enable Default value: { 1 }
1.3.6.1.4.1.13576.10.1.40.6.3	wioMax-Connections	R/W	The maximal count of modbus connections Default value: { 15 }
1.3.6.1.4.1.13576.10.1.40.6.4	wioConnection-Timeout	R/W	Timeout of the modbus connection Default value: { 600 }
1.3.6.1.4.1.13576.10.1.40.6.5	wioModbus-WatchdogMode	R/W	Mode of the modbus watchdog Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.6.6	wioModbus-WatchdogTime	R/W	Timeout of the modbus watchdog Default value: { 100 }
1.3.6.1.4.1.13576.10.1.40.6.7	wioFreeModbus-Sockets	R/W	Unused and free modbus connections Default value: { 15 }
1.3.6.1.4.1.13576.10.1.40.6.8	wioModbus-ConnectionTable	-	
1.3.6.1.4.1.13576.10.1.40.6.8.1	wioModbus-ConnectionEntry	-	
1.3.6.1.4.1.13576.10.1.40.6.8.1.1	wioModbus-ConnectionIndex	R/W	Index of modbus connection
1.3.6.1.4.1.13576.10.1.40.6.8.1.2	wioModbus-ConnectionIp	R/W	IP address of modbus connection
1.3.6.1.4.1.13576.10.1.40.6.8.1.3	wioModbus-ConnectionPort	R/W	Port of modbus connection

16.2.16 Ethernet IP Group

The Ethernet IP group contains information and settings for the controller's Ethernet IP.

Table 432: WAGO MIB – Ethernet IP Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.40.7.1	wioEthernetIpEnable	R/W	Enable/disable the port of the Ethernet IP server 0 = port of Ethernet IP server disable 1 = port of Ethernet IP server enable Default value: { 0 }
1.3.6.1.4.1.13576.10.1.40.7.2	wioEthernetIpVariables-InputCount	R/W	
1.3.6.1.4.1.13576.10.1.40.7.3	wioEthernetIpVariables-OutputCount	R/W	
1.3.6.1.4.1.13576.10.1.40.7.4	wioEthernetIpVariables-PlcInputCount	R/W	
1.3.6.1.4.1.13576.10.1.40.7.5	wioEthernetIpVariables-PlcInputOffset	R/W	
1.3.6.1.4.1.13576.10.1.40.7.6	wioEthernetIpVariables-PlcOutputCount	R/W	
1.3.6.1.4.1.13576.10.1.40.7.7	wioEthernetIpVariables-PlcOutputOffset	R/W	
1.3.6.1.4.1.13576.10.1.40.7.8	wioEthernetIpRunIdle-HeaderOrginatorToTarget	R/W	
1.3.6.1.4.1.13576.10.1.40.7.9	wioEthernetIpRunIdle-HeaderTargetToOrginator	R/W	

16.2.17 Process Image Group

The process image group contains a list of information about the terminals connected to the controller.

Table 433: WAGO MIB – Process Image Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.50.1	wioModulCount	R	Count of modules
1.3.6.1.4.1.13576.10.1.50.2	wioAnalogOutLength	R	Length of analog output process datas
1.3.6.1.4.1.13576.10.1.50.3	wioAnalogInLength	R	Length of analog input process datas
1.3.6.1.4.1.13576.10.1.50.4	wioDigitalOutLength	R	Length of digital output process datas
1.3.6.1.4.1.13576.10.1.50.5	wioDigitalInLength	R	Length of digital input process datas
1.3.6.1.4.1.13576.10.1.50.6	wioDigitalOutOffset	R	Offset of digital output process datas
1.3.6.1.4.1.13576.10.1.50.7	wioDigitalInOffset	R	Offset of digital input process datas
1.3.6.1.4.1.13576.10.1.50.8	wioModuleTable	-	
1.3.6.1.4.1.13576.10.1.50.8.1	wioModuleEntry	-	
1.3.6.1.4.1.13576.10.1.50.8.1.1	wioModuleNumber	R	Number of module slot
1.3.6.1.4.1.13576.10.1.50.8.1.2	wioModuleName	R	Name of module

Table 433: WAGO MIB – Process Image Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.50.8.1.3	wioModuleType	R	Type of module
1.3.6.1.4.1.13576.10.1.50.8.1.4	wioModuleCount	R	Count of module
1.3.6.1.4.1.13576.10.1.50.8.1.5	wioModule-AlternativeFormat	R	Module in alternative format
1.3.6.1.4.1.13576.10.1.50.8.1.6	wioModuleAnalog-OutLength	R	Length of analog output data of module (Bit)
1.3.6.1.4.1.13576.10.1.50.8.1.7	wioModuleAnalog-InLength	R	Length of analog input data of module (Bit)
1.3.6.1.4.1.13576.10.1.50.8.1.8	wioModuleDigital-OutLength	R	Length of digital output data of module (Bit)
1.3.6.1.4.1.13576.10.1.50.8.1.9	wioModuleDigital-InLength	R	Length of digital input data of module (Bit)

16.2.18 Plc Data Group

The Plc data group contains values that can be used for data exchange with CoDeSys.

Table 434: WAGO MIB – Plc Data Group

Identifier	Entry	Access	Description
1.3.6.1.4.1.13576.10.1.100.1	wioPlcDataTable	-	
1.3.6.1.4.1.13576.10.1.100.1.1	wioPlcDataEntry	-	
1.3.6.1.4.1.13576.10.1.100.1.1.1	wioPlcDataIndex	R/W	Number of plc data DWORD
1.3.6.1.4.1.13576.10.1.100.1.1.2	wioPlcDataReadArea	R/W	Readable plc data (DWORD)
1.3.6.1.4.1.13576.10.1.100.1.1.3	wioPlcDataWriteArea	R	Write-/readable plc data DWORD)

List of Figures

List of Tables

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: +49/5 71/8 87 – 0
Fax: +49/5 71/8 87 – 1 69
E-Mail: info@wago.com

Internet: http://www.wago.com