Manual

WAGO-I/O-SYSTEM 750
ETHERNET Fieldbus Coupler
750-352
10/100 Mbit/s; digital and analog signals

Version 1.0.0

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:
"ETHERNET Fieldbus Coupler" 750-352 of the WAGO-I/O-SYSTEM 750 series.

The ETHERNET Fieldbus Coupler 750-352 shall only be installed and operated according to the instructions in this manual and the system description for the WAGO-I/O-SYSTEM 750.

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

1.2   Copyright

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

1.3   Symbols

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

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

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

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

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

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

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

Information

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

1.4   Number Notation

Table 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.

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 fieldbus station 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).

3   System Description

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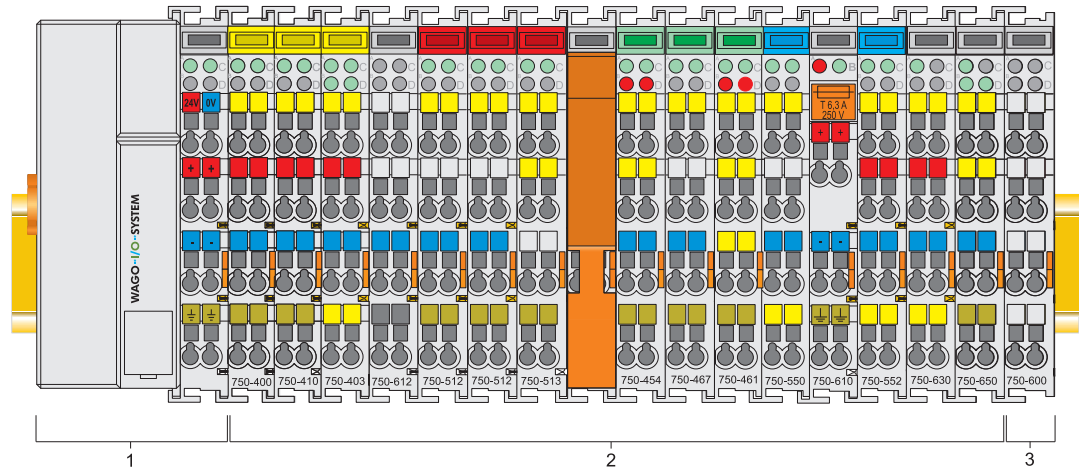
Figure 1: Field bus node

Couplers/controllers are available for different fieldbus systems.

The ECO coupler contains the fieldbus interface, electronics and a power supply for the system. 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.

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

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

The 3 wire technology supplemented by a ground wire connection allows for direct sensor/actuator wiring.

3.1   Manufacturing Number

The manufacturing number indicates the delivery status directly after production.

This number is part of the lateral marking on the component. In addition the manufacturing number is also printed on the cover of the configuration and programming interface of the fieldbus coupler or controller.
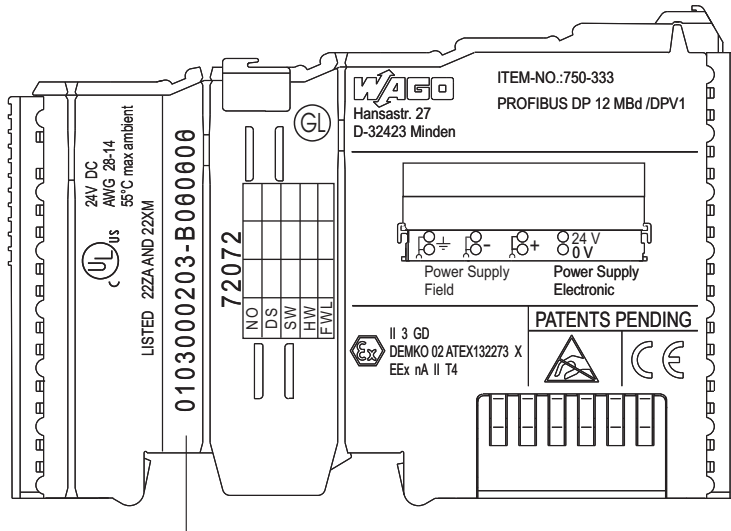
Manufacturing number

01	**03**	**00**	**02**	**03**	**-B000000**
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 ETHERNET Fieldbus Coupler 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)

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.
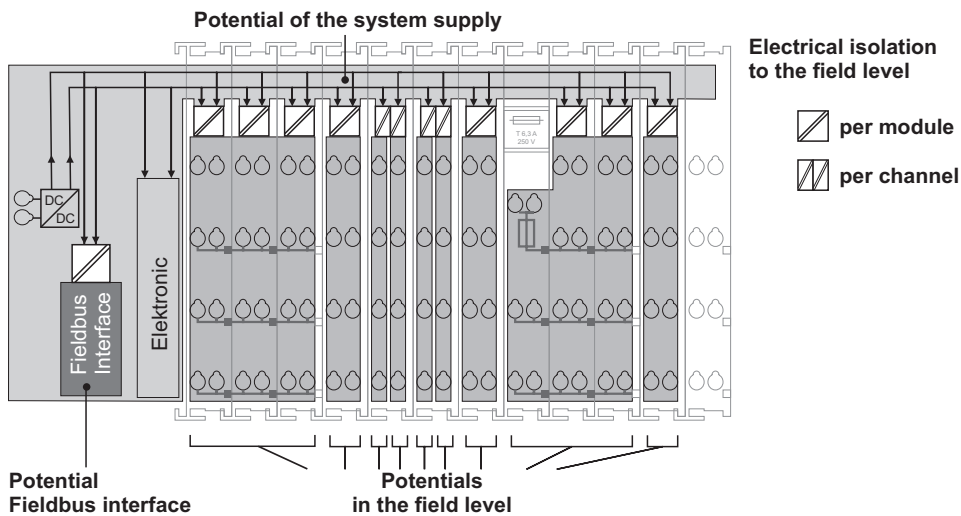
Figure 3: Isolation

> ## 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 (ring format, please see chapter "Grounding" > "Grounding Protection", Ring Feeding). Thus, if a bus module comes loose from a composite during servicing, then the protective conductor connection is still guaranteed for all connected field devices.
> When 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 (-15 % or +20 %). The power supply is provided via the coupler/controller and, if necessary, in addition via the internal system supply modules (750-613). The voltage supply is reverse voltage protected.

NOTICE

Do not use an incorrect voltage/frequency!
The use of an incorrect supply voltage or frequency can cause severe damage to the component.
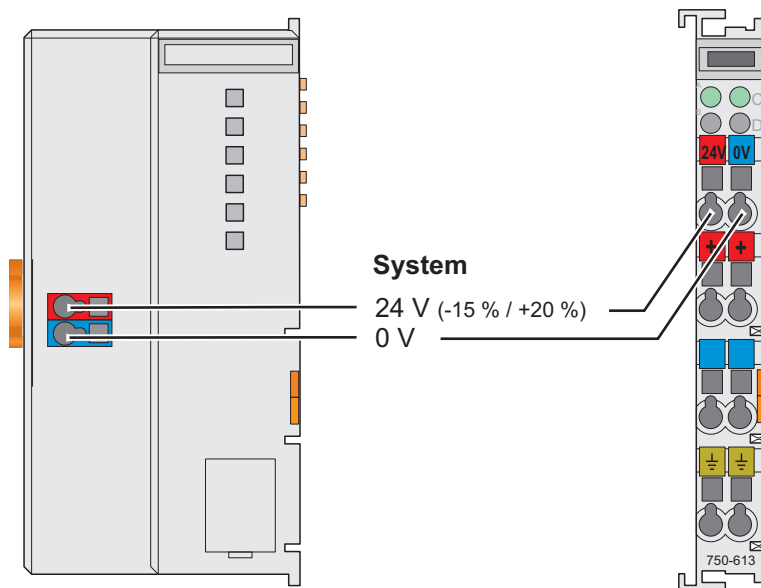
Figure 4: System supply

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.
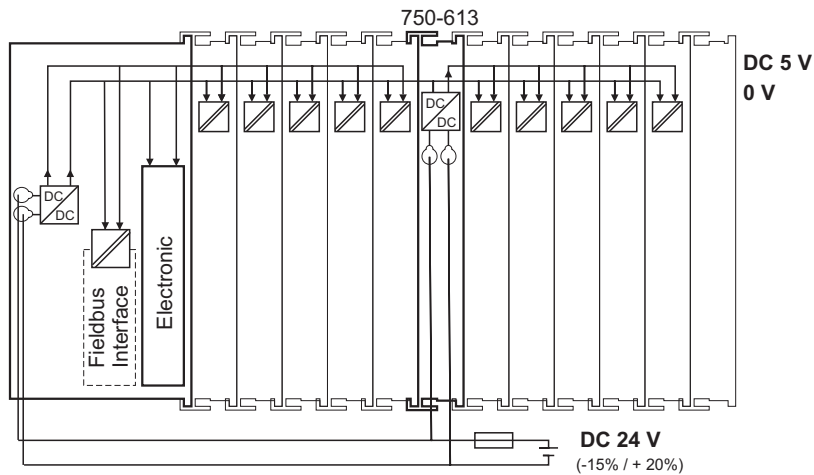
Figure 5: System voltage

> **Note**
>
> **Only reset the system simultaneously for all supply modules!**
> Resetting the system by switching on and off the system supply, must take place simultaneously for all supply modules (coupler/controller and 750 613).

3.6.2.2   Alignment

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

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

Table 3: Alignment

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

*) See current catalog, manuals, Internet

Example:

Calculating the current consumption on a 750-343 PROFIBUS DP ECO Coupler:

Internal current consumption	350 mA at 5 V
Residual current for bus modules	650 mA at 5 V
Sum I$_{(5\ V)\ total}$	**1000 mA at 5V**

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

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 an ECO Coupler:

A node with a PROFIBUS DP ECO coupler 750-343 consists of:
10 relay modules (750-517) and 20 digital input modules (750-405).

Internal current consumption	10 * 90 mA =	900 mA
	20 * 2 mA =	40 mA
Sum		**940 mA**

The PROFIBUS DP ECO coupler 750-343 can provide 650 mA for the bus modules. 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 accuracy check.

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

Coupler or controller

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

Internal system supply module 750-613

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

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

η = 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 aligned 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.

For the field side power, a power supply module is necessary.

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.

Further Supply modules
- DC 24 V
- AC/DC 0-230 V
- AC 120 V
- AC 230 V
- Fuse
- Diagnosis

Field supply

24 V
(-15 % / + 20 %)

0 V

Protective conductor

Power jumper contacts

Potential distribution to adjacent I/O modules
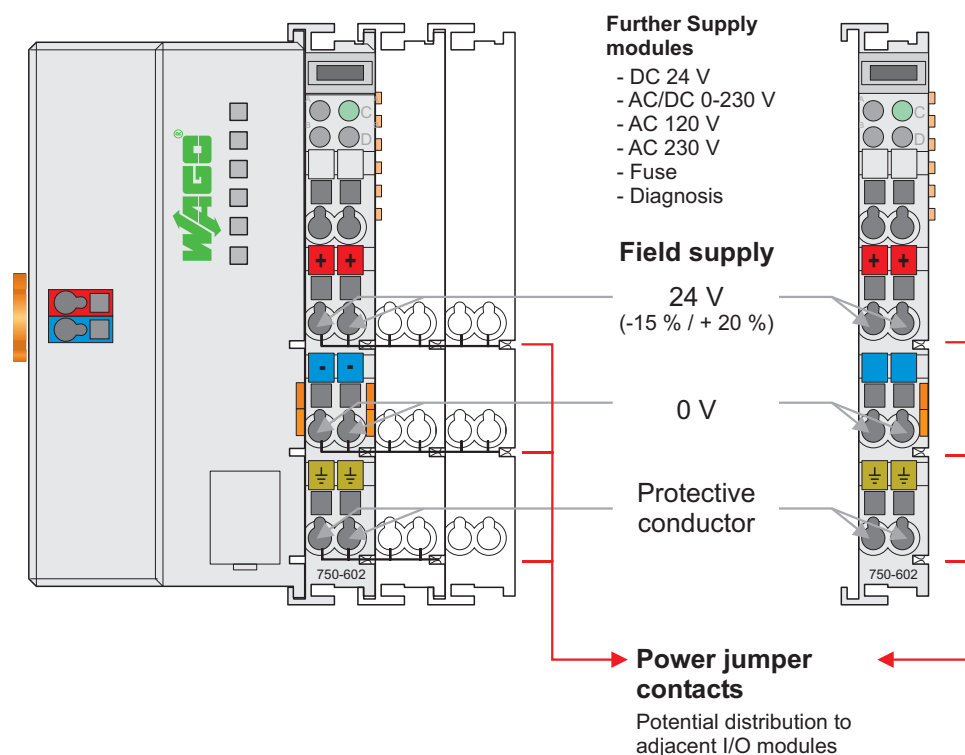
Figure 6: Field supply (sensor/actuator)

Note

→ **In exceptional instances, I/O modules can be directly connected to the field supply!**
The 24 V field supply can be connected also directly to a bus module, if the connection points are not needed for the peripheral device supply. In this case, the connection points need the connection to the power jumper contacts.

> ### *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

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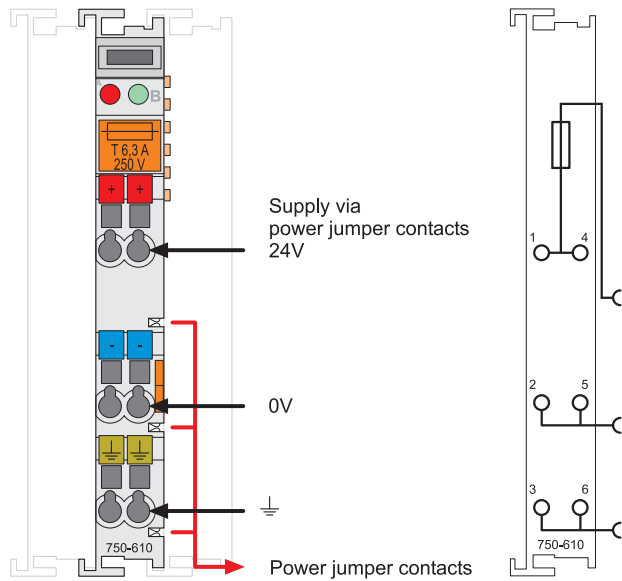
Figure 7: Supply module with fuse carrier (Example 750-610)

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.
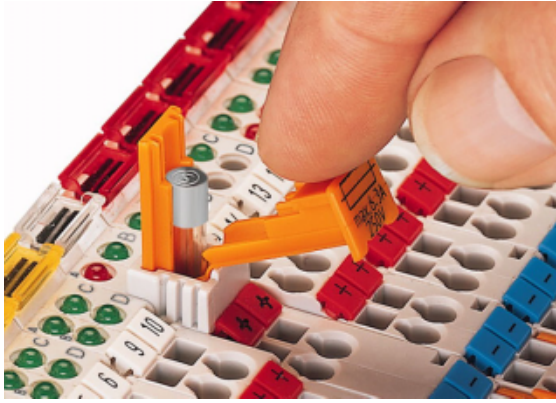
Figure 9: Opening the fuse carrier

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.
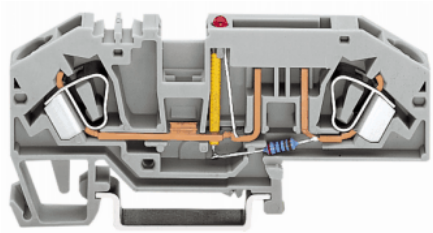
Figure 11: Fuse modules for automotive fuses, series 282
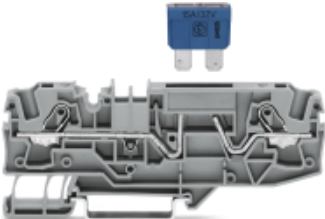
Figure 12: Fuse modules for automotive fuses, series 2006
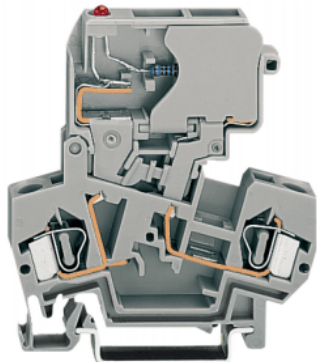
Figure 13: Fuse modules with pivotable fuse carrier, series 281
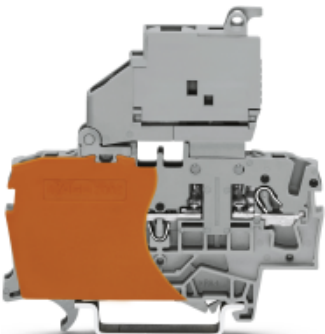
Figure 14: Fuse modules with pivotable fuse carrier, series 2002

3.6.4   Supply Example

→
Note

The system supply and the field supply shall be separated!
You should separate the system supply and the field supply in order to ensure bus operation in the event of a short-circuit on the actuator side.
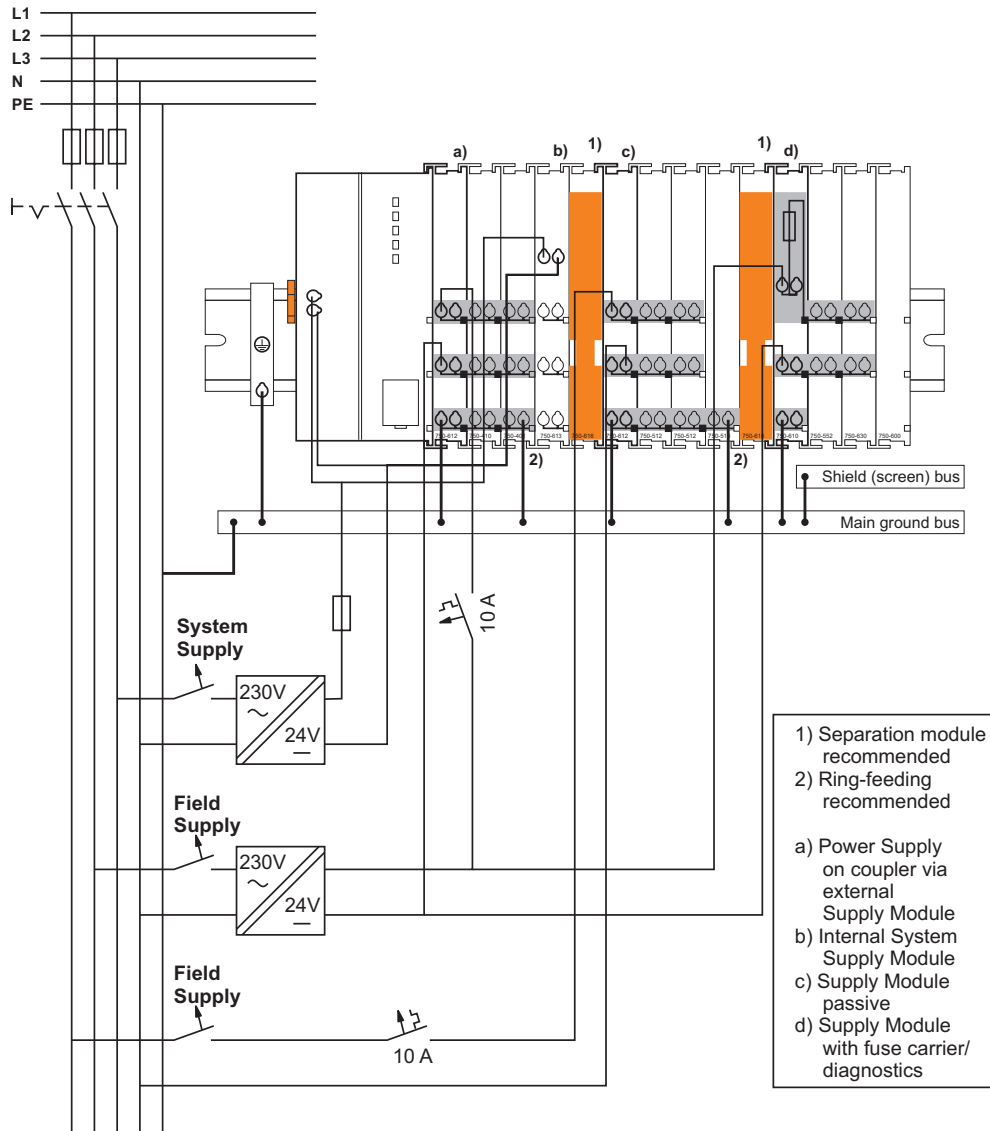
Figure 15: Supply example

3.6.5   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 IEC61131-2!**
> 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 5: 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 electronic 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 conduction connection 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 6: WAGO ground wire terminals

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

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.
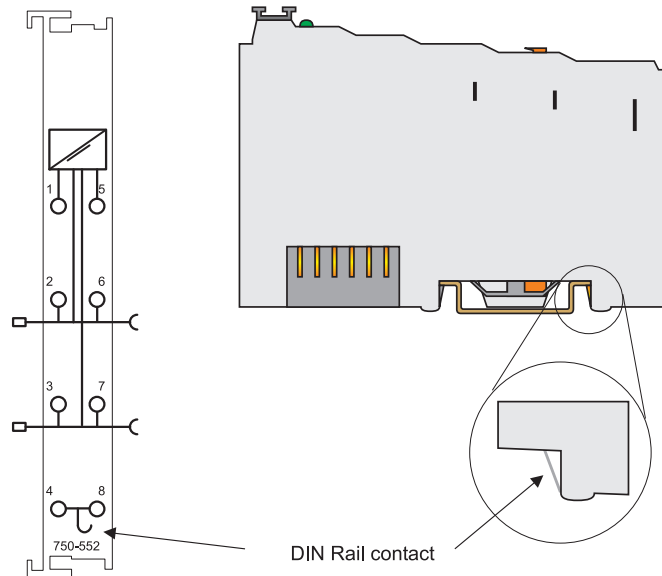
Figure 16: Carrier rail contact

⚠ 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, please see chapter "**Carrier Rail Properties**", page 56.

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.

Ring-feeding
of the ground
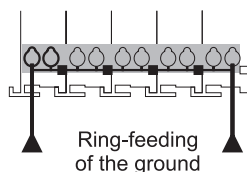
Figure 17: Ring-feeding

> **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 (Screening)

3.8.1   General

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

Note

Lay the shielding throughout the entrance and over a wide area!
Constant shielding is absolutely required in order to ensure the technical
specifications in terms of the measurement accuracy.
The cable shield should be potential. With this, incoming disturbances can be
easily diverted.
You should place shielding over the entrance of the cabinet or housing in order to
already repel disturbances at the entrance.

Note

Lay high-voltage cables separately!
Separate the data and signal conductors from all high-voltage cables.

3.8.2   Bus Conductors

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

3.8.3   Signal Conductors

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

Note

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

3.8.4   WAGO Shield (Screen) Connecting System

The WAGO Shield Connecting system includes a shield clamping saddle, a collection of rails and a variety of mounting feet. Together these allow many different possibilities. See catalog W4 volume 3 chapter 10.
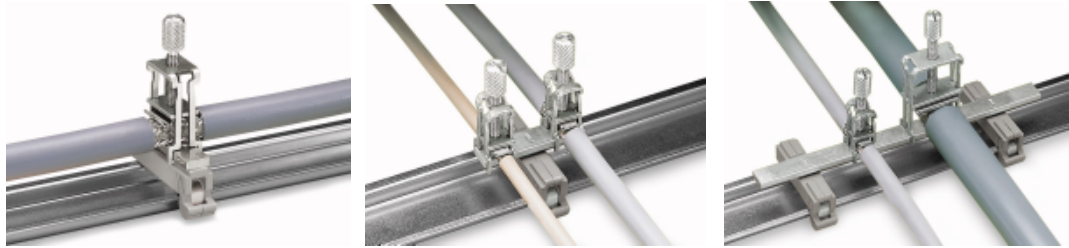
Figure 18: Example WAGO Shield (Screen) Connecting System
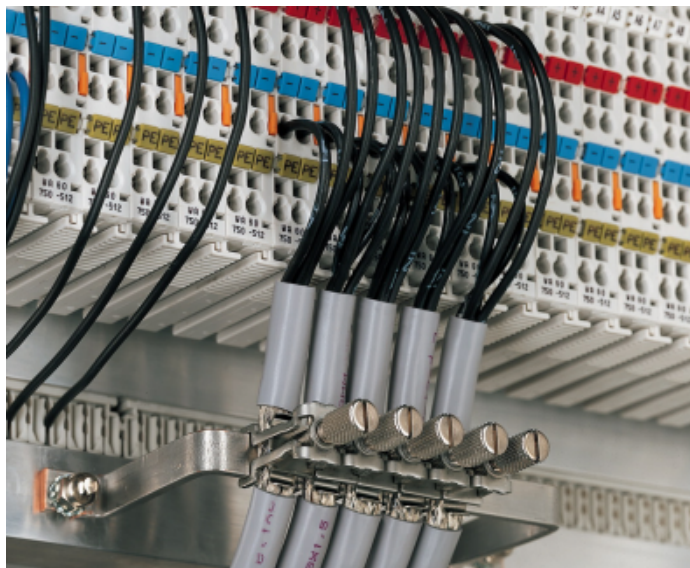
Figure 19: Application of the WAGO Shield (Screen) Connecting System

4   Device Description

The 750-352 Fieldbus Coupler connects the WAGO-I/O-SYSTEM 750 or 753 to the ETHERNET fieldbus system.

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

Equipped with two RJ-45 ports, which both work as 2-channel switches, the Fieldbus coupler enables easy and cost-effective cabling such as linear bus topology for which no additional external switches or hubs are required.

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 Coupler, all input signals from the sensors are combined. After connecting the Fieldbus Coupler, the Fieldbus Coupler 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 Coupler.

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 Coupler automatically begins a new word.

All sensor input signals are grouped in the coupler (slave) and transferred to the higher-order controller (master) via the fieldbus. Process data linking is performed in the higher-order controller. The higher-order controller puts out the resulting data to the actuators via the bus and the node.

The fieldbus connection consists of two ports (RJ-45). An ETHERNET switch integrated in the fieldbus coupler operates in the store and forward mode.
Both ports support:

- 10BASE-T / 100BASE-TX

- Full / Half duplex

- Autonegotiation

- Auto-MDI(X)

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

The MODBUS/TCP(UDP) protocol and the ETHERNET/IP protocol are implemented for exchanging process data.
The two communication protocols can be used optional or together.

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.

An internal server is available for Web-based applications.

HTML pages stored in the Fieldbus Coupler allow access to information about the configuration, the status and the I/O data of the fieldbus node via Web browsers. It is also possible to store individual HTML pages using the implemented file system, store custom HTML pages.

4.1   View

The view below shows the different parts of the device:

- The fieldbus connection is within the lower range on the left side.
- Over the fieldbus connection is a power supply unit for the system supply.
- LEDs for bus communication, error messages and diagnostics are within the upper range on the right side.
- Down right the service interface is to be found.
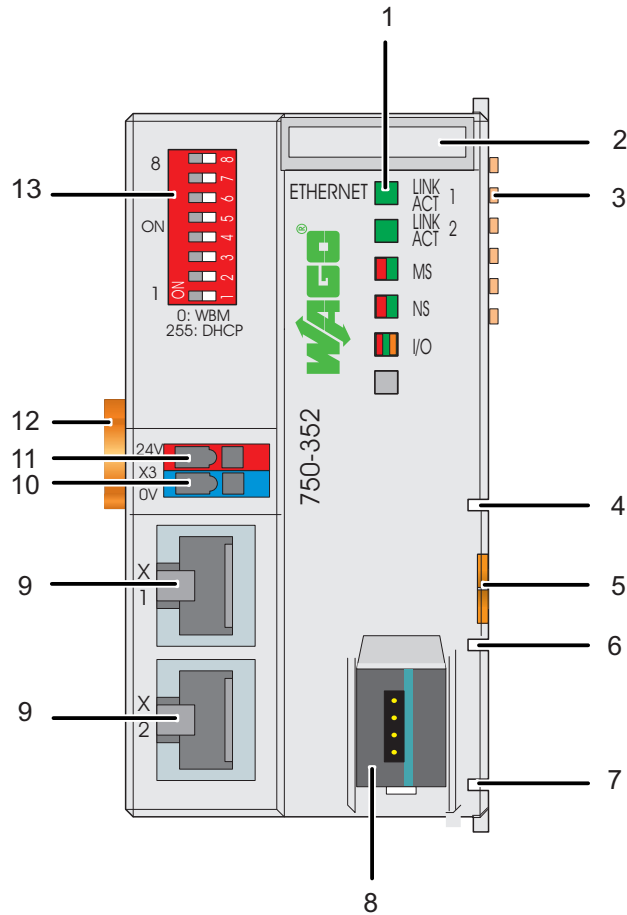
Figure 20: View ETHERNET TCP/IP Fieldbus Coupler

Table 7: Legend to the View ETHERNET TCP/IP Fieldbus Coupler

No.	Designa-tion	Meaning	Details see Chapter:
1	LINK ACT 1, 2, MS, NS, I/O	Status LEDs Fieldbus	„Device Description" > „Display Elements"
2	---	Marking possibility on four miniature WSB markers	---
3	---	Data Contacts	"Connect Devices" > "Data Contacts/Internal Bus"
4	---	Unlocking Lug	"Assembly" > "Inserting and Removing Devices"
5	---	Service Interface (open flap)	"Device Description" > "Operating Elements"
6	X1, X2	Fieldbus connection 2 x RJ-45 as 2-Port ETHERNET Switch	„Device Description" > „Connectors"
10	-	CAGE CLAMP® Connections Field Supply DC 0 V	"System Description" >"Voltage Supply"
11	+	CAGE CLAMP® Connections Field Supply DC 24 V	"System Description" >"Voltage Supply"
9	---	Locking Disc	„Assembly" > „Plugging and Removal of the Device"
10	---	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 via the transducer.
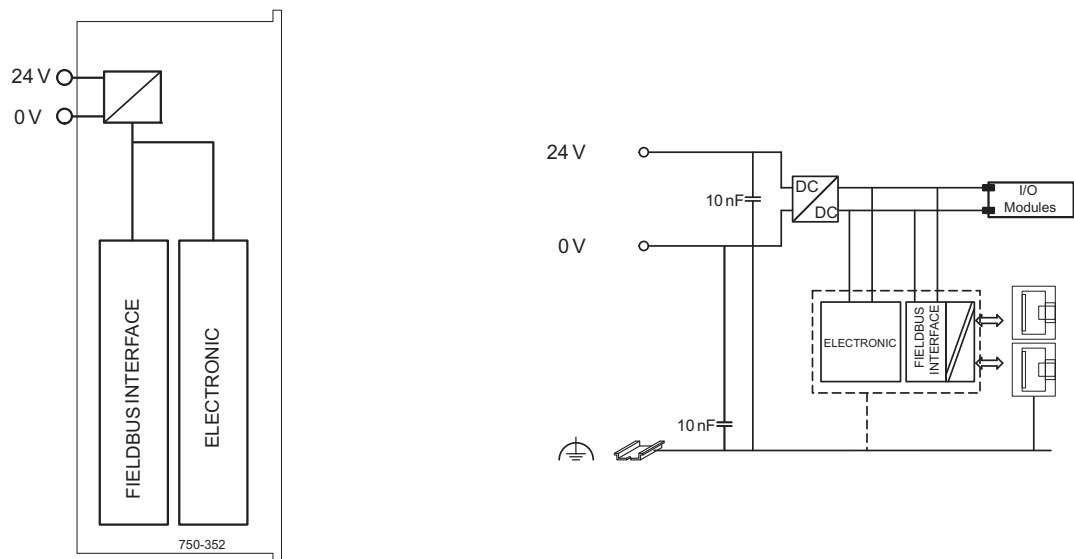
Figure 21: Device Supply

4.2.2   Fieldbus Connection
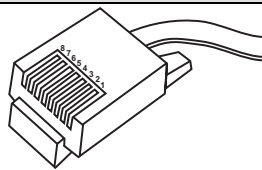
The connection to the fieldbus is made via two RJ-45 plugs, which are connected to the fieldbus controller via an integrated switch.
The integrated switch works in store-and-forward operation and for each port, supports the transmission speeds 10/100 Mbit as well as the transmission modes full and half-duplex and autonegotiation.
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 8: RJ-45 Connector and RJ-45 Connector Configuration

View	Contact	Signal	
Figure 22: 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 coupler 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).
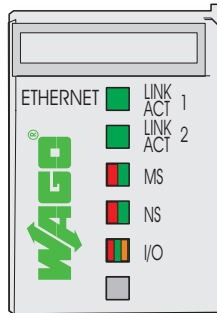
Figure 23: Display Elements

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

Table 9: 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 10: 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

> **ℹ** **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.

The configuration interface is used for the communication with the WAGO-I/O-*CHECK* and for downloading firmware.
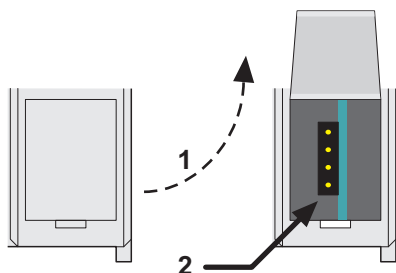
Figure 24: Service Interface for the configuration (closed and opened flap)

Table 11: Service Interface

Number	Description
1	Flap opened
2	Configuration 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 Communication Cable is connected to the 4-pole header.

4.4.2   **Address Selection Switch**
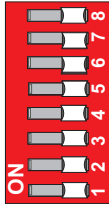
Figure 25: Address Selection Switch

The configuration of the IP address via the address selection switch takes place when you set the host ID (last digit of the IP address).

The coding of the host ID is bit by bit and begins with address selection switch 1 for bit 0 (LSB) and ends with address selection switch 8 for bit 7 (MSB).

The base address used depends on the IP address currently saved in the coupler. With the original factory settings, the IP address is configured to the value 0.0.0.0. by default. In this case, the static base address 192.168.1.X is used.

4.5   Technical Data

4.5.1   Device Data

Table 12: Technical Data - Device

Width	51 mm
High (from upper edge of DIN 35 rail)	65* mm (*from upper edge of DIN 35 rail)
Length	100 mm
Weight	160 g
Material	Polycarbonate, Polyamide 6.6
Installation	on DIN 35 rail with interlock
Modular interconnection	double feather key dovetail
Mounting position	any orientation
Marking	Standard marking label type group marking label 8 x 47 mm

4.5.2   Safe electrical Isolation

Table 2: Technical Data – Safe electrical Isolation

Air and creepage distance	Acc. to IEC 60664-1
Degree of pollution acc. to IEC 61131-2	2

4.5.3   Degree of protection

Table 3: Technical Data - Degree of protection

Degree of protection	IP 20

4.5.4   Supply

Table 4: Technical Data - Supply

Voltage Supply	DC 24 V (-25% ... +30%)
Input current$_{max.}$ (24 V)	25 mA
Efficiency of the power supply	87 %
Internal current consumption (5 V)	ca. 300 mA
Total current for I/O modules (5 V)	1700 mA
Isolation	500 V system/supply
Voltage via power jumper contacts	DC 24 V (-25 % ... +30 %)
Current via power jumper contacts max.	DC 10 A

4.5.5   Communication

Table 5: Technical Data - Communication

Number of fieldbus nodes per master	Limited by ETHERNET specification; max. 20 x 750-352 series connected
Transmission medium	Twisted Pair CAT 5e (S/UTP oder S/STP)
Fieldbus coupler connection	RJ-45
max. length of fieldbus segment	10 m between two participants, specified through ETHERNET 100Base-TX
Baud rate	100 Mbit/s
Protocols	SERCOS III, http, FTP, TCP/IP, UDP, SNTP, BootP, DHCP
Number of I/O modules with bus extension	64 250

Fieldbus
      Input process image max.
      Output process image max.
	
	2048 Bytes (RTC und SVC)
	2048 Bytes (RTC und SVC)

Configuration	via SERCOS III-Master
Communication profile	SERCOS III SCP_VarCFG
Device profile	SERCOS III GDP_Basic
I/O-Profile	SERCOS III FSP_IO Version 1.1.1

4.5.6   Wire Connection

Table 13: Technical Data Wire Connection

Wire connection	CAGE CLAMP®
Cross section	0.08 mm² ... 1.5 mm² / AWG 28-16
Stripped lengths	5 ... 6 mm / 0.22 in
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 14: Technical Data - Climatic environmental conditions

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

4.5.8   Mechanical strength

Table 15: 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" (Item-No.: 0888-0412) or via the internet
> under: www.wago.com → Service → Documentation → WAGO-I/O-SYSTEM
> 750 → System Description.

The following approvals have been granted to 750-352 fieldbus
coupler/controller:

cULus (UL508)

The following approvals are pending for 750-352 fieldbus coupler/controller:

CE   Conformity Marking

The following ship approvals are pending for 750-352 fieldbus coupler/controller:

GL (Germanischer Lloyd)   Cat. A, B, C, D (EMC 1)
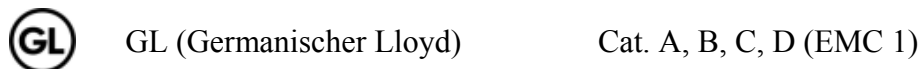
> ### *Information*
>
> **For more information about the ship approvals:**
> Note the "Supplementary Power Supply Regulations" chapter for the ship
> approvals.

4.7   Standards and Guidelines

750-352 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

750-352 meets the following requirements on emission and immunity of interference:

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

5.2   Total Extension

The length of the module assembly (including one end module of 12mm width)
that can be connected to the 750-352 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 750-352 must not exceed 780 mm.
Furthermore, you must observe restrictions made on certain types of
couplers/controllers (e.g. for PROFIBUS).

> ## 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 750-352, 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 16: WAGO DIN Rail

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

5.4   **Spacing**

The spacing between adjacent components, cable conduits, casing and frame sides must be maintained for the complete fieldbus node.
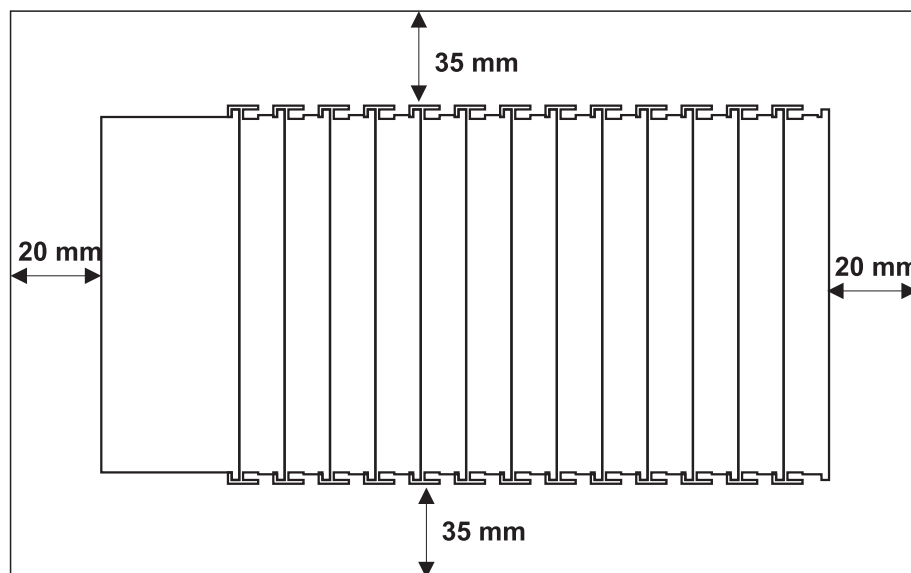
Figure 26: Spacing

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.
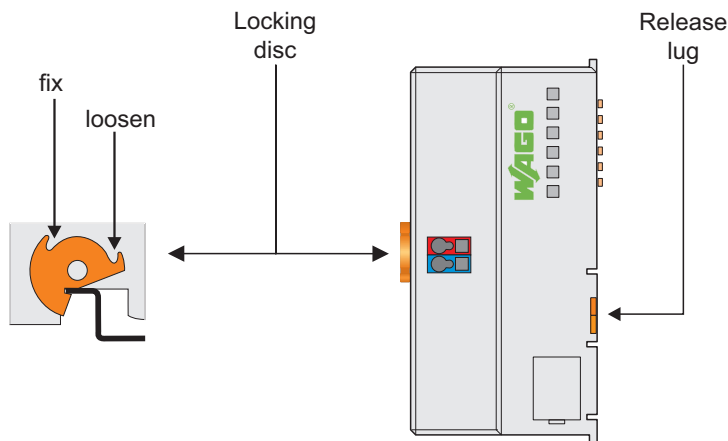
Figure 27: Unlocking lug

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 28: Insert I/O module

2. Press the I/O module into the assembly until the I/O module snaps into the carrier rail.
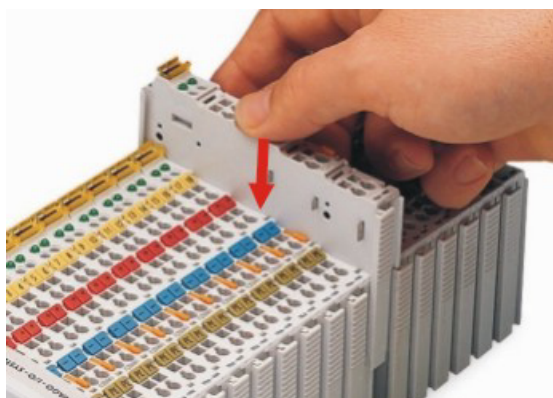
Figure 29: Snap the I/O module into place

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.
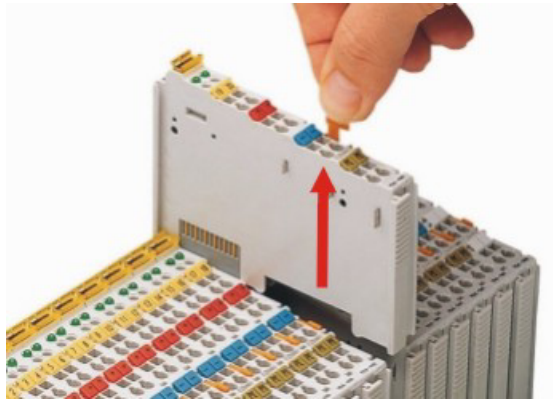
Figure 30: Removing the I/O module

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 31: 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.
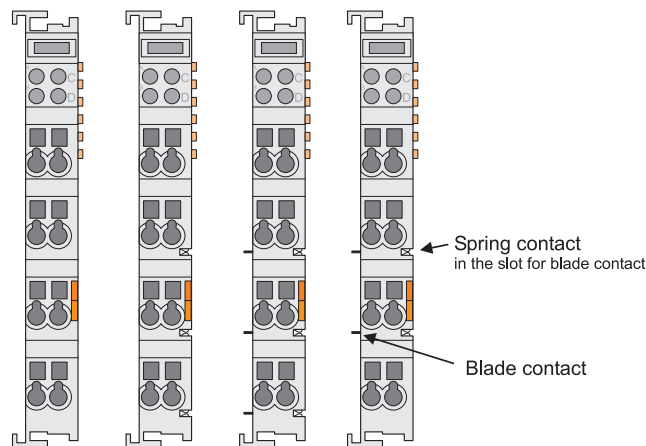
Figure 32: Example for the arrangement of power contacts

> **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.
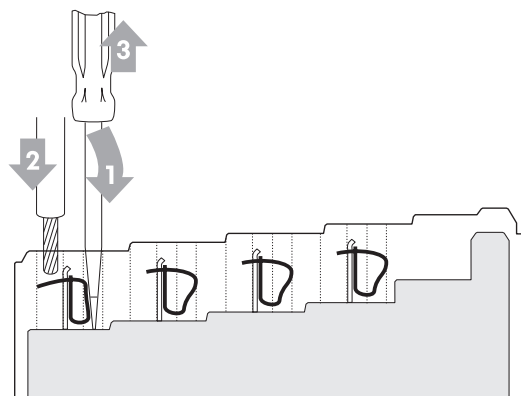
Figure 33: Connecting a conductor to a CAGE CLAMP®

7   Function Description

7.1   Operating System

After master configuration and electrical installation of the fieldbus station, the system is operative.

The coupler begins running up after switching on the power supply or after a reset.

Upon initialization, the fieldbus coupler determines the I/O modules and configuration. The 'I/O' LED flashes red. After a trouble-free start-up, the coupler enters "Fieldbus start" mode and the 'I/O' LED lights up green.
In the event of a failure, the 'I/O' LED will blink continuously. Detailed error messages are indicated by blinking codes; an error is indicated cyclically by up to 3 blinking sequences.
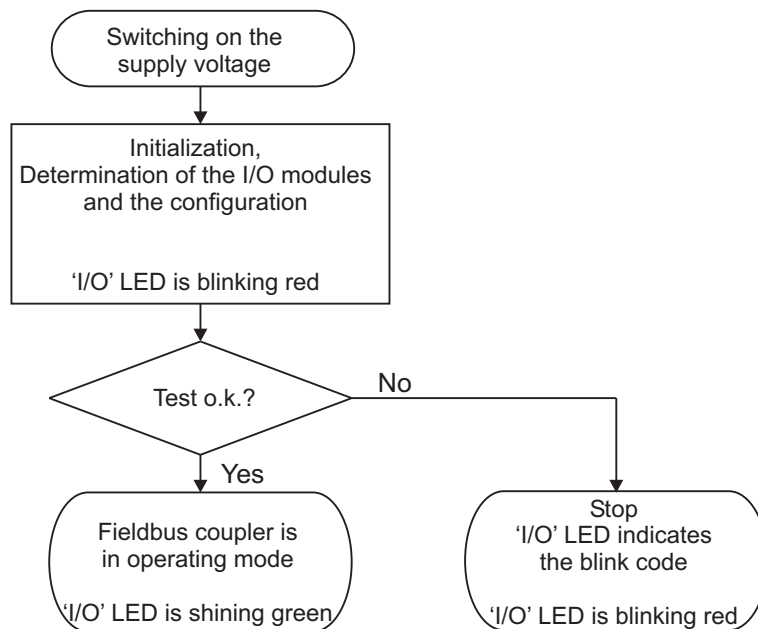
Figure 34: Operating System

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".

7.2   Process Data Architecture

After switching on the supply voltage, the fieldbus coupler identifies all I/O modules connected with the node that send or receive data (data width/bit width > 0). In the maximum total extension the node can consist of a mixed arrangement of a maximum of 64 analog and digital I/O modules, connected on the fieldbus coupler.

The data of the digital I/O modules are bit-oriented; i.e., digital data are sent bit by bit. The data of the analog I/O modules are byte-oriented; i.e., analog data are sent byte by byte. The term "Analog I/O modules" represents the group of byte-oriented I/O modules, which send data byte by byte. This group includes, e.g. counter modules, I/O modules for angle and distance measurement, and communication modules.

Table 17: Data width of the I/O Modules

Data width = 1 bit per Channel	Data width ≥ 1 word per Channel
Digital input modules	Analog input modules
Digital output modules	Analog output modules
Digital output modules with diagnostics	Analog input modules for Thermocouples
Power supply modules with diagnostics	Analog input modules for RTDs
Solid State Relay	Pulse width output modules
Relais output modules	Interface modules
Up/Down Counter	
I/O modules for angle and distance measurement	

The fieldbus coupler stores the process data in the process images. The fieldbus coupler works with a process output data image (PIO) and a process input data image (PII).

The PIO is filled of the fieldbus master with the process output data. The PII is filled of the fieldbus coupler with the process input data.

Into the input and output process image the data of the I/O modules are stored in the sequence of its position after the fieldbus coupler in the individual process image.

First, all the byte-oriented I/O modules are stored in the process image, then the bit-oriented I/O modules. The bits of the digital I/O modules are grouped into bytes. If the amount of digital I/O information exceeds 8 bits, the fieldbus coupler automatically starts a new byte.

NOTICE

Avoid equipment damages due to addressing errors!
To avoid equipment damages within the field range, you must consider that, 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.

Note

Consider the Process Data size for each module!
Observe the number of input and output bits or bytes for the individual I/O modules.

Note

Expandable with Module Bus Extension Coupler and End Module!
With the use of the WAGO Module Bus Extension Coupler Module 750-628 and the End Module 750-627, it is possible to operate up to 250 modules on the device.

For some I/O modules and their different versions, the structure of the process data depends on the fieldbus.

Information

Additional information about the fieldbus specific process image
For the fieldbus-specific process image of any WAGO-I/O-Module, please refer to the section "Structure of the Process Data".

7.3   Data Exchange

With the fieldbus coupler, data is exchanged either via the MODBUS/TCP protocol or via Ethernet/IP.

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

The fieldbus couplers of the WAGO-I/O-SYSTEM 750 are slave devices.

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 coupler 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 coupler)

-     15 connections via MODBUS/TCP (to read or write input and output data of the coupler)

-     128 connections for Ethernet IP

-     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 ETHERNET Fieldbus Coupler is essentially equipped with two interfaces for data exchange:

-     the interface to the fieldbus (Master)
-     the interface to the I/O modules.

Data exchange takes place between the fieldbus master and the I/O modules.

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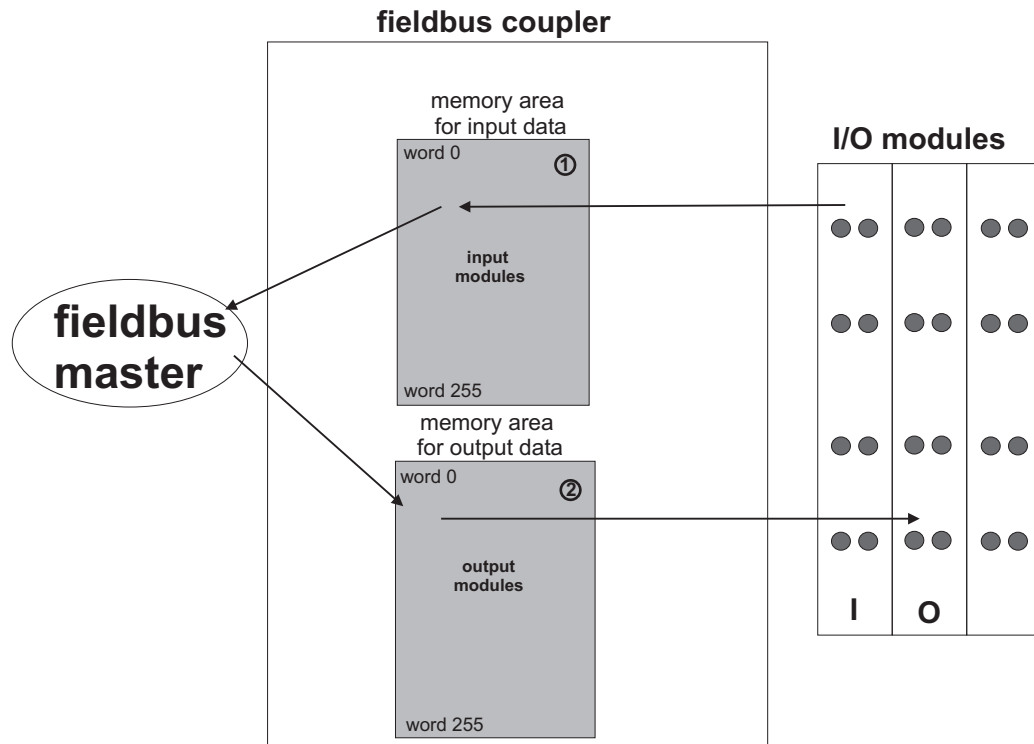
fieldbus coupler

A Figure 35: Memory areas and data exchange

The coupler 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.

In addition, all output data is mirrored in the ETHERNET Fieldbus Coupler 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.

7.3.1   Addressing

Module inputs and outputs in a fieldbus coupler 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.

7.3.1.1   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 18: 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	

7.3.1.2   Address Ranges

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

7.3.2   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 19: 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.
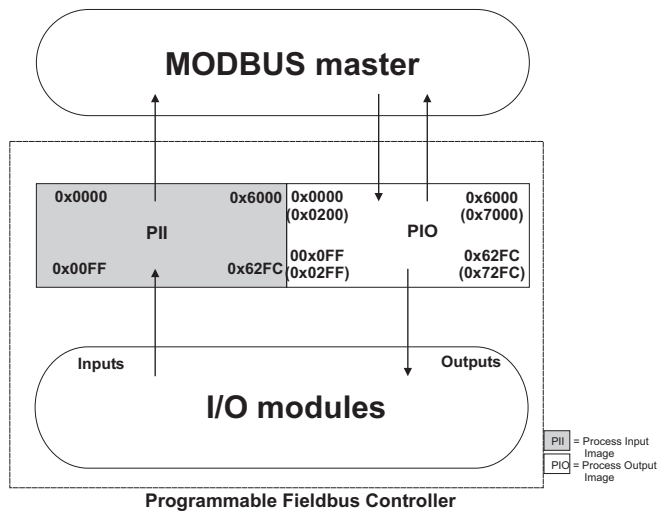
Figure 36: Data exchange between MODBUS Master and I/O modules

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".

7.3.2.1   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".

---

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**

- **Restoring factory settings**

8.1   Connecting Client PC and Fieldbus Nodes

1. Mount the fieldbus node on the TS 35 carrier rail.
    Follow the assembly instructions found in the "Assembly" chapter.

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

3. Connect the PC's Ethernet interface to the fieldbus coupler's Ethernet
    interface.

4. Turn the operating voltage on.

The fieldbus coupler is initialized. The coupler determines the I/O module
configuration and creates a process image.
During start-up, the I/O LED (red) flashes.
If the I/O LED lights up green after a brief period, the fieldbus coupler 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

- **Assigning IP Address via WAGO ETHERNET Settings**
  Assigning of IP addresses via the serial communication port

- 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 WAGO-ETHERNET-Settings

This program is used to configure an IP address via serial communications
interface, to reset the fieldbus coupler parameters to the factory settings and to
delete and extract the Flash File System in which the HTML pages of the fieldbus
coupler are stored.
WAGO communication cables or WAGO radio-link adapters can be used for data
communication.

NOTICE

Do not connect 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

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

1.   Using a WAGO Communication cable 750-920, connect the serial interface
      of your PC with the service interface of the fieldbus coupler.

2.   Start **WAGO-ETHERNET-Settings** program.

3.   Click on **[Read]** to read in and identify the connected fieldbus node.

4.   Select the **TCP/IP** tab:

Figure 37: Setting IP addresses in WAGO-ETHERNET-Settings

5.   To assign a fixed address, select the "Use following addresses:" option for
      address assignment (BootP is the default).

6.   Enter the required **IP Address** and, if applicable, the address of the subnet
      mask and gateway.

7.   Click on the **[Write]** button to apply the address in the fieldbus node.

8.   You can now close ETHERNET-Settings or make other changes in the
      Web-based Management System as required. Click on **[Call WBM]**.

8.2.2   Assigning IP Address via Address Selection Switch

Use the address selection switch to set the host ID, i.e., the last byte ("X") of the IP address saved in the fieldbus coupler with values between 1 and 254 binary coded.

Example:

IP address saved in the fieldbus coupler:	192.168.7.33
Set DIP switch value:	**50** (binary coded: 00110010)
Resulting IP address:	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 ("X") 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 base address stored in the fieldbus coupler and the host ID set on the DIP switch.
The IP address make via the Web-based Management System or WAGO-ETHERNET-Settings is disabled.

Note

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

The base address used consists of the first three bytes of the IP address. This always depends on the IP address currently saved in the fieldbus coupler.
If there is still no static IP address in the fieldbus coupler, the default value **192.168.1.X** 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 "X".

Information

More information about changing the static base address
You can also change the base address currently saved in the fieldbus coupler as required.
Either proceed as described in the previous section "Assign IP Address Using WAGO-ETHERNET Settings" or 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).
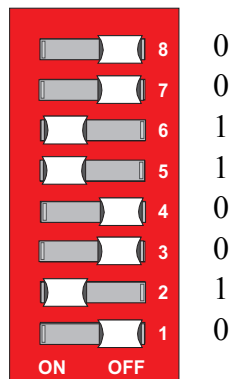
Figure 38: Address selection switch

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

8.2.3   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

8.2.3.1   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.

Note

DHCP must be enabled on the Web pages or in WAGO-ETHERNET-Settings (for this, address selection switch must be on 0)!
Note that DHCP must be enabled on the internal Web pages of the WBM, HTML page "Port configuration".
If BootP is also enabled, it is automatically disabled.
DHCP is disabled and BootP is enabled by default when delivered.
As an alternative, you can also enable DHCP via WAGO-ETHERNET-Settings in the **TCP/IP** tab.

An IP address is automatically assigned after restarting the fieldbus node.

Information

More information about reading the IP address
You can use WAGO-ETHERNET-Settings to read the IP address currently assigned.
Proceed as described in the section "Assigning IP Address via WAGO-ETHERNET-Settings."

8.2.3.2   Disabling DHCP

Note

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

You can disable DHCP in three ways:

- Disable DHCP via the address selection switch.

- Disable DHCP in WAGO-ETHERNET-Settings.

- 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 enables DHCP and disables the DIP switch.

1. Use the address selection switch to set a value between 1 and 254. The
   address saved in the fieldbus coupler is then valid (with changed host ID =
   DIP switch).
   (Example: If the address 10.127.3.15 was saved in the fieldbus coupler and
   you set the switch to 50 (binary coded 00110010), for example, the fieldbus
   coupler 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 WAGO-ETHERNET-Settings

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!

1. Set the address selection switch to 0.

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

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

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

The fieldbus coupler is initialized. The coupler determines the I/O module
configuration and creates a process image.
During start-up, the I/O LED (red) flashes.
If the I/O LED lights up green after a brief period, the fieldbus coupler 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 **WAGO-ETHERNET-Settings** program.

5.   Click on **[Read]** to read in and identify the connected fieldbus node.

6.   Select the **TCP/IP** tab:

Figure 39: Setting IP addresses in WAGO-ETHERNET-Settings

7.   To assign a fixed address, select the "Use following addresses:" option for
      address assignment (BootP is the default).

8.   Click on the **[Write]** button to apply the address in the fieldbus node.

9.   You can now close ETHERNET-Settings or make other changes in the
      Web-based Management System as required. Click on **[Call WBM]**.

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. 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 based Management System loads.

3. Select "Port" 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 "Port configuration" loads:

Figure 40: WBM page "Port"

5. Disable DHCP by selecting the option "**BootP**" or "**use IP fom EEPROM**".

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.

8.2.4   Assigning the IP Address with a BootP Server

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

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

Note

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

The following steps are included:

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

8.2.4.1   Note MAC ID

1.   Write down the couplers's MAC address (see label or peel-off strip).
     If the fieldbus is already installed, turn off the operating voltage of the
     fieldbus coupler, then take the fieldbus coupler out of the assembly of your
     fieldbus node and note the MAC ID of your fieldbus coupler.
     The MAC ID is applied to the back of the fieldbus coupler or on the self-
     adhesive peel-off strip on the side of the fieldbus coupler.

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

2.   Plug the fieldbus coupler 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 coupler (DC 24 V power supply unit).

The fieldbus coupler is initialized. The coupler determines the I/O module
configuration and creates a process image.
During start-up, the I/O LED (red) flashes.
If the I/O LED lights up green after a brief period, the fieldbus coupler 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.

---

i   *Information*

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

Error code 6, followed by error argument 4, is indicated by the I/O LED after
coupler 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.

8.2.4.2   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: _ _ _ . _ _ _ . _ _ _ . _ _ _

8.2.4.3   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).

8.2.4.4   Disabling BootP

When the BootP protocol is activated the coupler 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 coupler uses the configured 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 coupler, BootP must be disabled.
This prevents the fieldbus coupler from receiving a new BootP request.

Note

No loss if the IP Address with deactivated BootP Protocol!
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 BootP in two ways:

•   Disable BootP in **WAGO-ETHERNET-Settings**
•   Disable BootP in the **Web-based Management System**

Disable BootP in WAGO-ETHERNET-Settings

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!

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

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

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

The fieldbus coupler is initialized. The coupler determines the I/O module configuration and creates a process image.
During start-up, the I/O LED (red) flashes.
If the I/O LED lights up green after a brief period, the fieldbus coupler 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

i

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

Error code 6, followed by error argument 4, is indicated by the I/O LED after coupler 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.

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

5.   Click on **[Read]** to read in and identify the connected fieldbus node.

6.   Select the **TCP/IP** tab:

Figure 41: Setting IP addresses in WAGO-ETHERNET-Settings

7.   To assign a fixed address, select the "Use following addresses:" option for
      address assignment (BootP is the default).

8.   Click on the **[Write]** button to apply the address in the fieldbus node.

9.   You can now close ETHERNET-Settings or make other changes in the
      Web-based Management System as required. Click on **[Call WBM]**.

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 coupler. You can navigate to other information using the hyperlinks in
the left navigation bar.

Figure 42: 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 coupler IP can be changed in the network by the DHCP server!**
If BootP is not deactivated and an ISDN/DSL router is installed in the network (factory default setting with DHCP server activated) addresses will be assigned automatically from the address range for the ISDN/DSL router after a loss of power (loss of 24 V DC power to coupler). As a result, all couplers will be assigned new IP addresses!

---

4.   In the left navigation bar click on **Port** to open the HTML page for selecting a protocol.
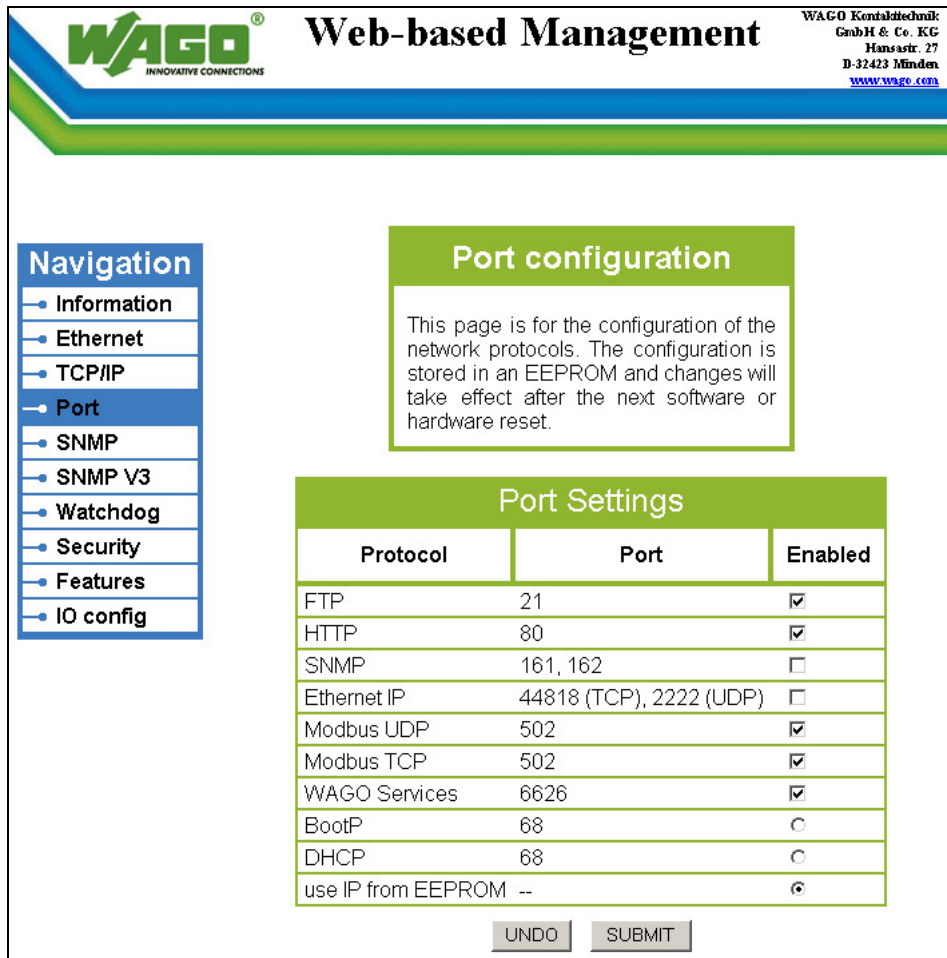
Figure 43: WBM page "Port"

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

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.

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

8.2.4.5   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

Information

More information about reading the IP address
You can use WAGO-ETHERNET-Settings to read the IP address currently assigned.
Proceed as described in the section "Assigning IP Address via WAGO-ETHERNET-Settings."

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.

The fieldbus coupler is initialized. The coupler determines the I/O module configuration and creates a process image.
During start-up, the I/O LED (red) flashes.
If the I/O LED lights up green after a brief period, the fieldbus coupler 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)
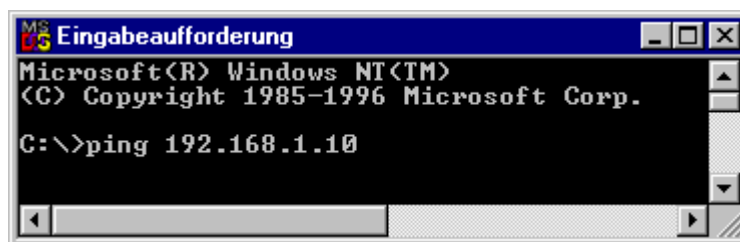
Figure 44: Example for the Function test of a Fieldbus Node

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 coupler 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 coupler 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 coupler.

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

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

The fieldbus coupler is initialized. The coupler determines the I/O module configuration and creates a process image.
During start-up, the I/O LED (red) flashes.
If the I/O LED lights up green after a brief period, the fieldbus coupler 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   Restoring Factory Settings

To restore the factory settings, proceed as follows:

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

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

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

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   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.

9.1   Information

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

Figure 45: WBM page "Information"

Table 20: WBM page "Information"

Coupler details			
Entry	**Default**	**Value (example)**	**Description**
Order number	750-352/000-000	750-352/000-000	Item 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)
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

9.2   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 46: WBM page "Ethernet"

Table 21: 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

Misc. Configuration				
Entry	**Port**			**Description**
	1	**2**	**internal**	
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.
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.

9.3   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 47: WBM page "TCP/IP"

Table 22: 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

9.4 Port

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

Figure 48: WBM page "Port"

Table 23: WBM page "Port"

Port Settings				
Entry	**Entry**		**Entry**	
FTP (Port 21)	Enabled	☑	☑	activating "File Transfer Protocol"
			☐	deactivating "File Transfer 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
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!

WAGO®

9.5 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 coupler 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.

9.5.1　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 49: WBM page "SNMP"

Table 24: WBM page "SNMP"

SNMP Configuration		
Entry	**Value (Default)**	**Description**
Name of device	750-352	Device name (sysName)
Description	ETHERNET Fieldbus Coupler 750-352	Device description (sysDescription)
Physical location	LOCAL	Location of device (sysLocation)
Contact	support@wago.com	E-mail contact address (sysContact)

SNMP v1/v2 Manager Configuration			
Entry	**Value (Default)**		**Description**
Protocol Enable	SNMP V1/V2c	☑	☑ Activating SNMP Version 1/2c
			☐ Deactivating SNMP-Version 1/2c
Local Community Name	public		Used community name

SNMP v1/v2 Trap Receiver Configuration			
Entry	**Value (Default)**		**Description**
Trap Receiver 1	0.0.0.0		IP address of **1.** used SNMP manager
Community Name 1	public		**1.** Community name of the network community used
Trap Version	V1	◉	V1 ◉ V2 ○ Activating Traps Version 1
	V2	○	V1 ○ V2 ◉ Activating Traps Version 2
Trap Receiver 2	0.0.0.0		IP address of **2.** used SNMP manager
Community Name 2	public		**2.** Community name of the network community used
Trap Version	V1	◉	V1 ◉ V2 ○ Activating Traps Version 1
	V2	○	V1 ○ V2 ◉ Activating Traps Version 2

9.5.2   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 50: 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).

9.6   Watchdog

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

Figure 51: WBM page „Watchdog"

Table 25: 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)

9.7   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 52: WBM page "Security"

Table 26: 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

9.8   Features

Use the "Features" HTML page to enable or disable additional functions.
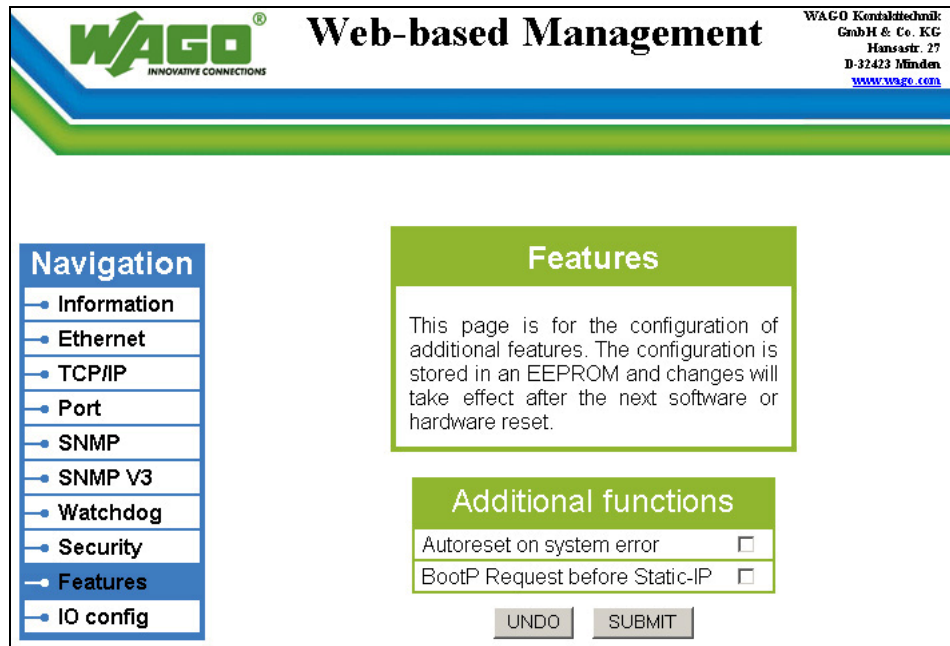
Figure 53: WBM page "Features"

Table 27: WBM page "Features"

Entry	Default	Description
Additional functions		
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.

9.9   I/O Config

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

The node structure created using the "WAGO-I/O-*PRO* CAA I/O Configurator" hardware configuration tool is displayed in the window. If no modules are shown in this window, no hardware configuration and, thus, no allocation of write access privileges have been assigned. In this case, the handling defined at the Web site "PLC" by the function "I/O configuration - Compatible handling for ea-config.xml" will be applied to assign the write privileges for all outputs either to the standard fieldbus, or to the PLC.
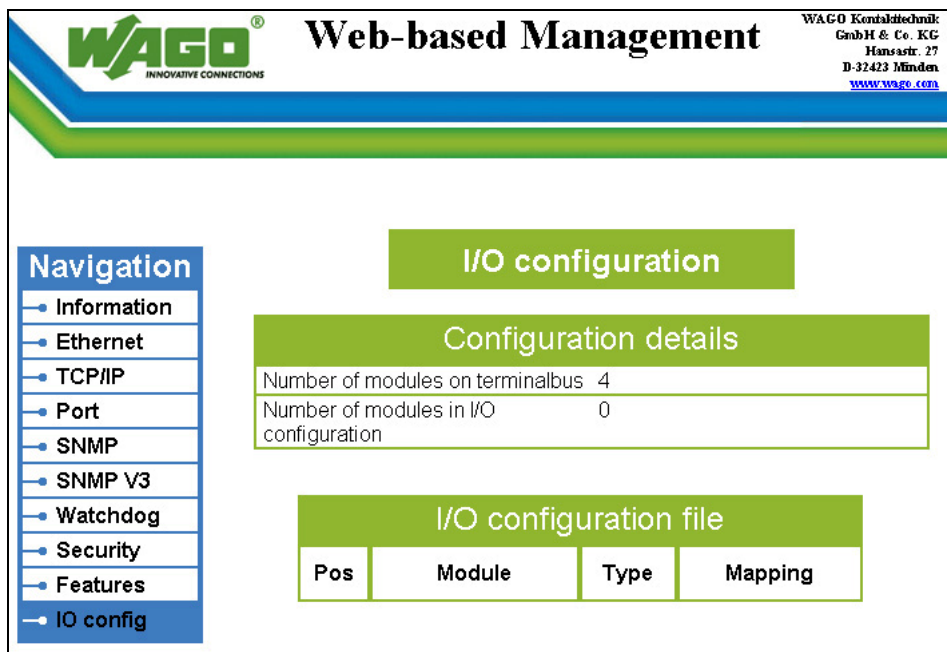
Figure 54: WBM page "IO config"

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

Table 28: 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 Configurator (see the following note)

10   Diagnostics

10.1   LED Signaling

For on-site diagnostics, the fieldbus coupler has several LEDs that indicate the operational status of the coupler or the entire node (see following figure).
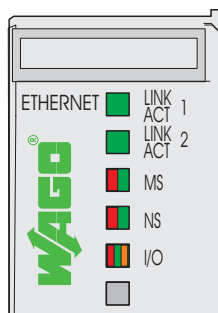
Figure 55: Display Elements

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 29: LED assignment for diagnostics

Diagnostics area	LEDs
Fieldbus status	• LINK ACT Port 1 • LINK ACT Port 2 • MS • NS
Node status	• I/O

10.1.1   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 30: 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 by BootP, DHCP or the Ethernet Settings tool.

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

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

Table 31: 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.
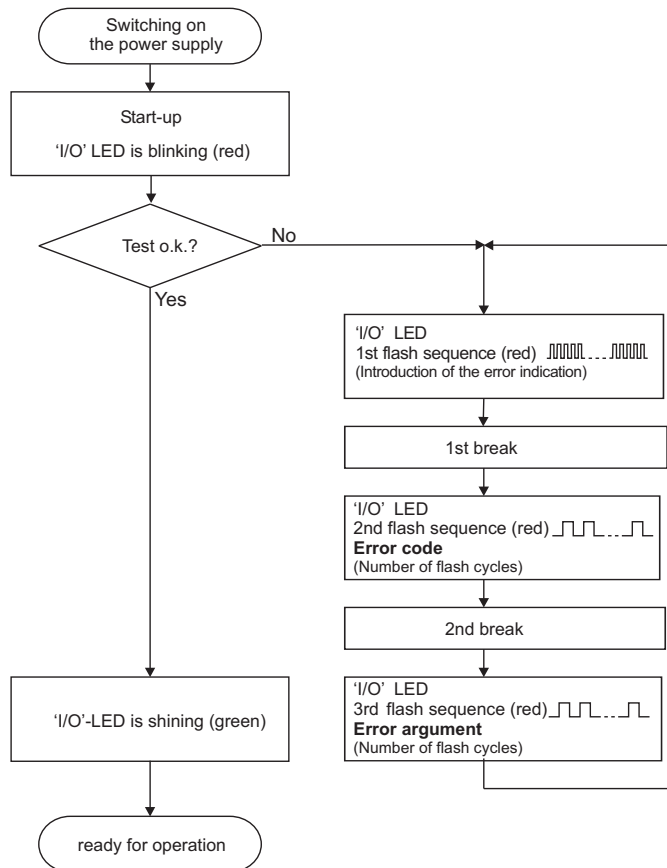
Figure 56: Node status - I/O LED signaling
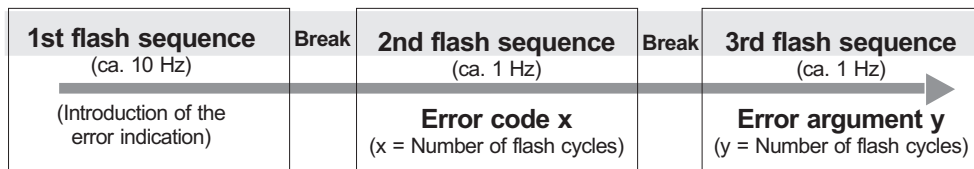
Figure 57: Error message coding

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 32: 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 coupler.
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 coupler.
3	Invalid check sum in the parameter area of the fieldbus coupler.	1. Turn off the power supply for the node. 2. Replace the fieldbus coupler and 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 coupler and 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 coupler and turn the power supply on again.
6	The I/O module configuration after AUTORESET differs from the configuration determined the last time the fieldbus coupler was powered up.	1. Restart the fieldbus coupler 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 coupler and turn the power supply on again.

Table 32: 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 coupler and turn the power supply on again.
9	Bus controller initialization error	1. Turn off the power supply for the node. 2. Replace the fieldbus coupler and turn the power supply on again.
10 ... 13	not used	
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 33: 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 34: 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 coupler). - 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 coupler). 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 coupler.

Table 35: 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 coupler	1. Turn off the power supply to the node. 2. Plug in an end module behind the fieldbus coupler. 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 coupler. - 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 coupler). - 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 coupler). 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 coupler and the LED is flashing, either the I/O module or fieldbus coupler 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 36: 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 37: 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 coupler. 3. Turn the power supply on again.
2	Ethernet Hardware initialization error	1. Restart the fieldbus coupler by turning the power supply off and on again. 2. If the error still exists, exchange the fieldbus coupler.
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 coupler

10.2   Fault Behavior

10.2.1   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.

```
┌─────────────────────────────────────┐
│   FBUS_ERROR_INFORMATION             │
│                                      │
│                                      │
│              FBUS_ERROR  ├──         │
│                  ERROR   ├──         │
└─────────────────────────────────────┘
```
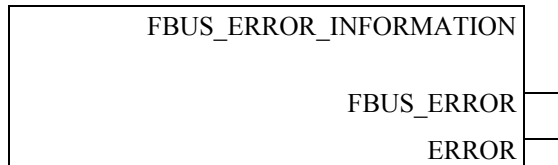
Figure 58: Function block for determining loss of fieldbus, independently of protocol

'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 CAA. 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

10.2.2   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 coupler 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 coupler 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

11   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.

11.1   Implemented Protocols

11.1.1   Communication Protocols

11.1.1.1   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 38: 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 39: Network Class A

e. g.	101	.	16	.	232	.	22
01100101		00010000		11101000		00010110	
0	Net-ID			Host-ID			

The highest bit in Class A networks is always '0'. This means the highest byte can be in a range of '0 0000000' to '0 1111111'.
Therefore, the address range of a Class A network in the first byte is always between 0 and 127.

- **Class B:** (Net-ID: Byte 1 … Byte 2, Host-ID: Byte 3… Byte 4)

Table 40: Network Class B

e. g.	181	.	16	.	232	.	22
10110101		00010000		11101000		00010110	
10	Net-ID			Host-ID			

The highest bits in Class B networks are always '10'. This means the highest byte can be in a range of '10 000000' to '10 111111'.
Therefore, the address range of Class B networks in the first byte is always between 128 and 191.

- **Class C:** (Net-ID: Byte 1 … Byte 3, Host-ID: Byte 4)

Table 41: Network Class C

e. g.	201	.	16	.	232	.	22
11000101		00010000		11101000		00010110	
110	Net-ID					Host-ID	

The highest bits in Class C networks are always '110'. This means the highest byte can be in a range of '110 00000' to '110 11111'.

Therefore, the address range of Class C networks in the first byte is always between 192 and 223.

- **Additional network classes (D, E):** are only used for special tasks.

Key Data

Table 42: Key Data Class A, B and C

Network Class	Address range of the subnetwork	Possible number of	
		Networks	Hosts per Network
Class A	1.XXX.XXX.XXX ... 126.XXX.XXX.XXX	127 (2^7)	Approx. 16 Million (2^{24})
Class B	128.000.XXX.XXX ... 191.255.XXX.XXX	Approx. 16 Thousand (2^{14})	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 (programmable) fieldbus coupler or controller can be easily assigned an IP address via the implemented BootP protocol. For small internal networks we recommend selecting a network address from Class C.

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 43: 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 44: Subnet mask for Class A network

255	.0	.0	.0

- **Class B Subnet mask:**

Table 45: Subnet mask for Class B network

255	.255	.0	.0

- **Class C Subnet mask:**

Table 46: 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 47: 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.

11.1.1.2   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.

11.1.1.3   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.

11.1.2   Configuration and Diagnostics Protocols

11.1.2.1   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.

i

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 48: 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.

11.1.2.2   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 49: 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. IP address of the Network Time Server. When assigning an NTP server, the SNTP client is automatically enabled in the coupler.
[OPT42] NTP server	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.
[OPT51] Lease time	The renewing time indicates when the fieldbus coupler/controller must renew the lease time. The rebinding time should be approximately 7/8 of the lease time.
[OPT58] Renewing time	The rebinding time indicates after what amount of time the fieldbus coupler/controller must have received its new address.
[OPT59] Rebinding time	The renewing time should be approximately half of the lease time.

*) In contrast to BootP, the DHCP client does not support assignment of the host name.

11.1.2.3   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.

11.1.2.4   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.

11.1.2.5   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.

11.1.2.6   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.

11.1.2.6.1 MIB II Description

MIB II acc. to RFC1213 is divided into the following groups:

Table 50: 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..

11.1.2.6.2 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 51: 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

11.1.3   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.

11.2   MODBUS Functions

11.2.1   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 52: 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 53: 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 54: List of the MODBUS Functions in the Fieldbus Coupler

Function code		Function	Access method and description	Access to resources	
FC1	0x01	Read Coils	Reading of several single input bits	R:	Process image
FC2	0x02	Read Input Discretes	Reading of several input bits	R:	Process image
FC3	0x03	Read Multiple Registers	Reading of several input registers	R:	Process image, internal variables
FC4	0x04	Read Input Registers	Reading of several input registers	R:	Process image, internal variables
FC5	0x05	Write Coil	Writing of an individual output bit	W:	Process image
FC6	0x06	Write Single Register	Writing of an individual output register	W:	Process image, internal variables
FC11	0x0B	Get Comm Event Counters	Communication event counter	R:	None
FC15	0x0F	Force Multiple Coils	Writing of several output bits	W:	Process image
FC16	0x10	Write Multiple Registers	Writing of several output registers	W:	Process image, internal variables
FC22	0x16	Mask Write Register		W:	Process image
FC23	0x17	Read/Write Registers	Reading and writing of several output registers	R/W: Process image	

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.

11.2.2   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.
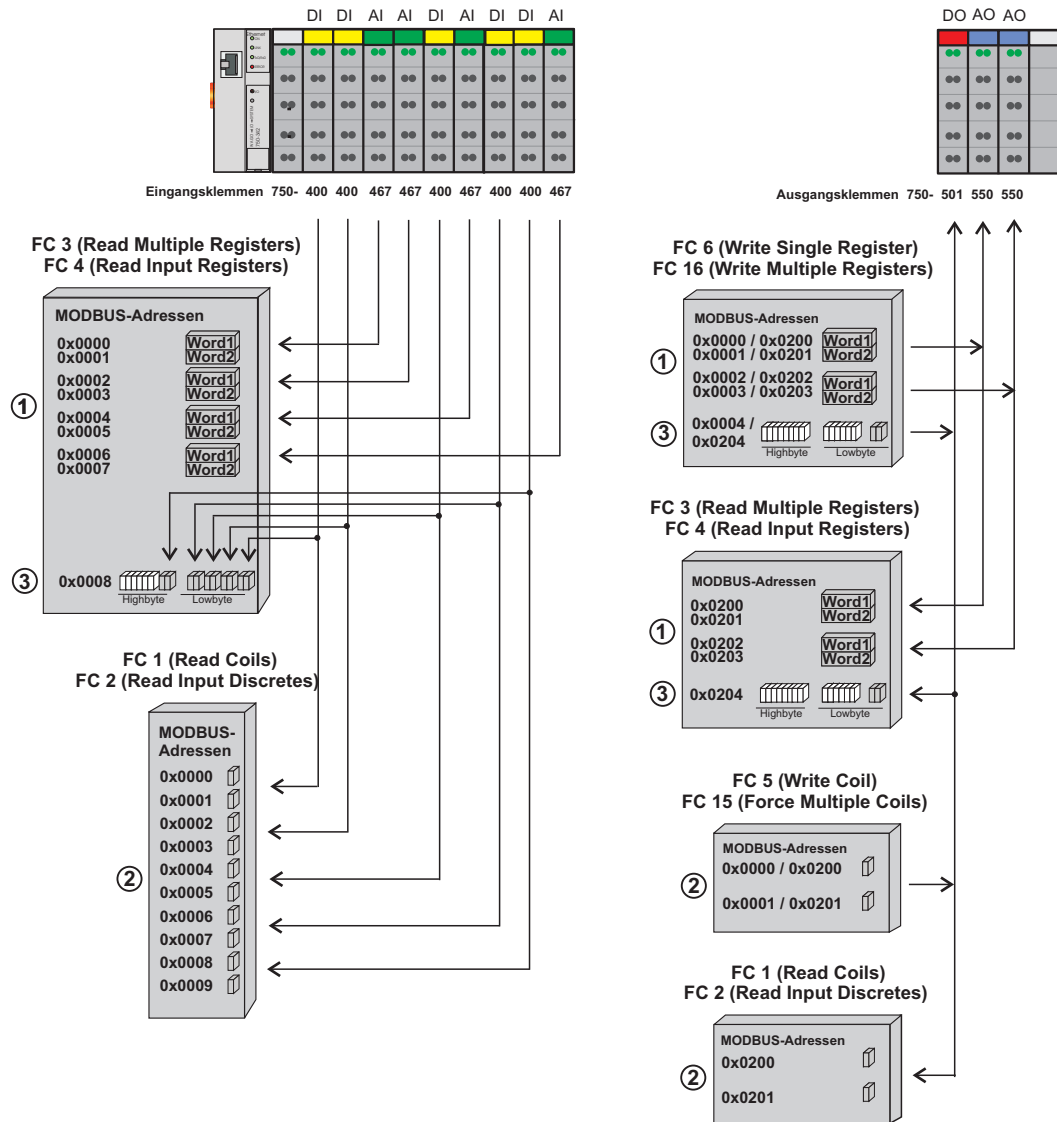
Figure 59: Use of the MODBUS Functions

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.

11.2.3   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 55: Exception odes

Exception code	Meaning
0x01	Illegal function
0x02	Illegal data address
0x03	Illegal data value
0x04	Slave device failure
0x05	Acknowledge
0x06	Server busy
0x08	Memory parity error
0x0A	Gateway path unavailable
0x0B	Gateway target device failed to respond

The following chapters describe the datagram architecture of request, response and exception with examples for each function code.

→ **Note**

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 $[6000_{hex} ... 62FC_{hex}]$.

11.2.3.1 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 56: 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 57: 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 58: 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 59: Exception of Function code FC1

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x81
Byte 8	Exception code	0x01 or 0x02

11.2.3.2 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 60: 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 61: 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 62: 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 63: Exception of Function code FC2

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x82
Byte 8	Exception code	0x01 or 0x02

11.2.3.3 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 64: 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 65: 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 66: Exception of Function code FC3

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x83
Byte 8	Exception code	0x01 or 0x02

11.2.3.4 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 67: 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 68: 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 69: Exception of Function code FC4

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x84
Byte 8	Exception code	0x01 or 0x02

11.2.3.5 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 70: 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 71: 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 72: Exception of Function code FC5

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01, 0x02 or 0x03

11.2.3.6   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 73: 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 74: 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 75: Exception of Function code FC6

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

11.2.3.7   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 76: 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 77: 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 78: Exception of Function code FC 11

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

11.2.3.8   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 79: 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 80: 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 81: Exception of Function code FC15

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x8F
Byte 8	Exception code	0x01 or 0x02

11.2.3.9   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 82: 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 83: 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 84: Exception of Function code FC16

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

11.2.3.10 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 85: 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 86: 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 87: Exception of Function code FC22

Byte	Field name	Example
...		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 or 0x02

11.2.3.11 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 88: 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 89: 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 90: 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.

11.2.4   MODBUS Register Mapping

The following tables display the MODBUS addressing and the internal variables.

Via the register services the states of the complex and digital I/O modules can be determined or changed.

Register Access Reading (with FC3 and FC4)

Table 91: Register Access Reading (with FC3 and FC4)

MODBUS address		IEC 61131	Memory range
[dec]	[hex]	address	
0...255	0x0000...0x00FF	%IW0...%IW255	Physical input area (1) First 256 Words of physical input data
256...511	0x0100...0x01FF	-	MODBUS exception: "Illegal data address"
512...767	0x0200...0x02FF	%QW0...%QW255	Physical output area (1) First 256 Words of physical output data
768...4095	0x0300...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...12287	0x1000...0x2FFF	-	Configuration register (see Section „Configuration Register")
12288...24575	0x3000...0x5FFF	-	MODBUS exception: "Illegal data address"
24576...25339	0x6000...0x62FB	%IW256...%IW1020	Physical input area (2) Additional 764 Words physical input data
25340...28671	0x62FC...0x6FFF	-	MODBUS exception: "Illegal data address"
28672...29435	0x7000...0x72FB	%QW256...%QW1020	Physical output area (2) Additional 764 Words physical output data
29436...65535	0x72FC...0xFFFF	-	MODBUS exception: "Illegal data address"

Register Access Writing (with FC6 and FC16)

Table 92: Register Access Writing (with FC6 and FC16)

MODBUS address		IEC 61131	Memory range
[dec]	[hex]	address	
0...255	0x0000...0x00FF	%QW0...%QW255	Physical output area (1) First 256 Words of physical output data
256...511	0x0100...0x01FF	-	MODBUS exception: "Illegal data address"
512...767	0x0200...0x02FF	%QW0...%QW255	Physical output area (1) First 256 Words of physical output data
768...4095	0x0300...0x0FFF	-	MODBUS exception: "Illegal data address"
4096...12287	0x1000...0x2FFF	-	Configuration register (see Section "Configuration Register")
12288...24575	0x3000...0x5FFF	-	MODBUS exception: "Illegal data address"
24576...25339	0x6000...0x62FB	%QW256...%QW1020	Physical output area (2) Additional 764 Words physical output data
25340...28671	0x62FC...0x6FFF	-	MODBUS exception: "Illegal data address"
28672...29435	0x7000...0x72FB	%QW256...%QW1020	Physical output area (2) Additional 764 Words physical output data
29436...65535	0x72FC...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 93: 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... 12287	0x0400...0x2FFF	-	MODBUS exception: "Illegal data address"
12288...13815	0x3000...0x35F7		MODBUS exception: "Illegal data address
13816...16383	0x35F8...0x3FFF	-	MODBUS exception: "Illegal data address"
16384...17911	0x4000...0x45F7		MODBUS exception: "Illegal data address
17912...32767	0x45F8...0x7FFF	-	MODBUS exception: "Illegal data address"
	0x8000…0x85F7	Physical input area (2)	Starts with the 513th and ends with the 2039th digital input
	0x85F8…0x8FFF		MODBUS exception: "Illegal data address"
	0x9000…0x95F7	Physical output area (2)	Starts with the 513th and ends with the 2039th digital output
	0x95F8…0xFFFF		MODBUS exception: "Illegal data address"

Bit Access Writing (with FC5 and FC15)

Table 94: Bit Access Writing (with FC5 and FC15)

MODBUS address		Memory range	Description
[dec]	[hex]		
0...511	0x0000...0x01FF	Physical output area (1)	First 512 digital outputs
512...1023	0x0200...0x03FF	Physical output area (1)	First 512 digital outputs
1024...12287	0x0400...0x2FFF	-	MODBUS exception: "Illegal data address"
12288...13815	0x3000...0x35F7		MODBUS exception: "Illegal data address
13816...16383	0x35F8...0x3FFF	-	MODBUS exception: "Illegal data address"
16384...17911	0x4000...0x45F7		MODBUS exception: "Illegal data address
17912...32767	0x45F8...0x7FFF	-	MODBUS exception: "Illegal data address"
	0x8000…0x85F7	Physical output area (2)	Starts with the 513th and ends with the 2039th digital output
	0x85F8…0x8FFF		MODBUS exception: "Illegal data address"
	0x9000…0x95F7	Physical output area (2)	Starts with the 513th and ends with the 2039th digital output
	0x95F8…0xFFFF		MODBUS exception: "Illegal data address"

11.2.5   MODBUS Registers

Table 95: 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
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 96: 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

11.2.5.1 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.

11.2.5.2 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. Finally, the Watchdog-Trigger register (0x1003) must be changed to a non-zero value to start the timer.

Reading the Minimum Trigger time (Register 0x1004) reveals whether a watchdog fault occurred. If this time value is 0, a fieldbus failure is assumed. The timer of watchdog can manually be reset, if it is not timed out, by writing a value of 0x1 to the Restart Watchdog register (0x1007).

After the watchdog is started, it can be stopped by the user via the Watchdog Stop register (0x1005) or the Simply Stop Watchdog register (0x1008).

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 97: 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 98: 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 0 or 1 FC 4 Bit 2 FC 5 Bit 0 or 2 FC 6 Bit 1 or 2 etc. The watchdog function is started if a value is not equal to zero. If only codes from non-supported functions are entered in the mask, the watchdog will not start. An existing fault is reset and writing into the process illustration is possible. Also here changes cannot be made while the watchdog is running. When the watchdog is enabled, no code is generated to rewrite the current data value.

Table 99: 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. These codes are currently not supported, for this reason the default value should not be changed. It is not possible to modify this value while the watchdog is running.

Table 100: 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. A watchdog fault is reset and writing process data is possible again.

Table 101: 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 102: 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 103: 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 104: 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 105: 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 106: 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 107: 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 108: 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 109: 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.

11.2.5.3   Diagnostic Registers

The following registers can be read to determine errors in the node:

Table 110: Register address 0x1020

Register address 0x1020 (4128$_{dec}$)	
Value	LedErrCode
Access	Read
Description	Declaration of the Error code

Table 111: Register address 0x1021

Register address 0x1021 (4129$_{dec}$)	
Value	LedErrArg
Access	Read
Description	Declaration of the Error argument

11.2.5.4   Configuration Registers

The following registers contain configuration information of the connected modules:

Table 112: 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 113: 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 114: Register address 0x1024

Register address 0x1024 (4132$_{dec}$)	
Value	CnfLen.DigitalOut
Access	Read
Description	Number of digital output bits in the process image

Table 115: Register address 0x1025

Register address 0x1025 (4133$_{dec}$)	
Value	CnfLen.DigitalInp
Access	Read
Description	Number of digital input bits in the process image

Table 116: Register address 0x1028

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

Table 117: 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 118: 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 119: 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 120: 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 121: 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 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 122: Register address 0x1050

Register address 0x1050 (4176$_{dec}$) with a word count of 3	
Value	Diagnosis of the connected I/O modules
Access	Read
Description	Diagnosis of the connected I/O modules, length 3 words Word 1: Number of the module Word 2: Number of the channel Word 3: Diagnosis

Table 123: 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 item numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 → Input module Bit position 1 → Output module Bit position 2…7 → Not used Bit position 8…14 → Module size in bits Bit position 15 → Designation digital module

Examples:

4 Channel Digital Input Module = 0x8401

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Code	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
Hex	8				4				0				1			

2 Channel Digital Output Module = 0x8202

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Code	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
Hex	8				2				0				2			

Table 124: Register address 0x2031

Register address 0x2031 (8241dec) 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 item numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 $\rightarrow$ Input module Bit position 1 $\rightarrow$ Output module Bit position 2…7 $\rightarrow$ Not used Bit position 8…14 $\rightarrow$ Module size in bits Bit position 15 $\rightarrow$ Designation digital module

Table 125: Register address 0x2032

Register address 0x2032 (8242dec) 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 item numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 $\rightarrow$ Input module Bit position 1 $\rightarrow$ Output module Bit position 2…7 $\rightarrow$ Not used Bit position 8…14 $\rightarrow$ Module size in bits Bit position 15 $\rightarrow$ Designation digital module

Table 126: Register address 0x2033

Register address 0x2033 (8243dec) 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 item numbers cannot be read out of digital modules, a code is displayed for them, as defined below: Bit position 0 $\rightarrow$ Input module Bit position 1 $\rightarrow$ Output module Bit position 2…7 $\rightarrow$ Not used Bit position 8…14 $\rightarrow$ Module size in bits Bit position 15 $\rightarrow$ Designation digital module

Table 127: Register address 0x2040

Register address 0x2040 (8256dec)	
Value	Implement a software reset
Access	Write (Write sequence 0xAA55 or 0x55AA)
Description	With Writing 0xAA55 or 0x55AA the register will be reset.

Table 128: 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 129: 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 130: Register address 0x2043

Register address 0x2043 (8259$_{dez}$)	
Value	0x55AA
Access	Write
Description	Factory Settings

11.2.5.5 Firmware Information Registers

The following registers contain information on the firmware of the controller:

Table 131: 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 132: 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 133: Register address 0x2012

Register address 0x2012 (8210$_{dec}$) with a word count of 1	
Value	Item number, INFO_ITEM
Access	Read
Description	WAGO item number, e.g. 841 for the controller 750-841 or 341 for the coupler 750-341 etc.

Table 134: 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 135: 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 136: 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

WAGO®

Table 137: 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 138: 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 139: 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

11.2.5.6 Constant Registers

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

Table 140: Register address 0x2000

Register address 0x2000 (8192$_{dec}$)	
Value	Zero, GP_ZERO
Access	Read
Description	Constant with zeros

Table 141: 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 142: 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 143: 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 144: 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 145: 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 146: 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 147: 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 148: 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

11.3   EtherNet/IP (Ethernet/Industrial Protocol)

11.3.1   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.

11.3.2   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 149: ISO/OSI reference model

7 Application Layer	Object Library (Communications, Applications, Time Synchronization)			Safety Object Library	Comon Industrial Protocol (CIP)
6 Presentation Layer	Data Management Services Explicit and I/O Messages			Safety Services and Messages	
5 Session Layer	Connection Management, Routing				
4 Transport Layer	TCP/UDP	CompoNet Network and Transport	ControlNet Network and Transport	DeviceNet Network and Transport	Network Adaptations of CIP
3 Network Layer	Internet Protocol				
2 Data Link Layer	Ethernet CSMA/CD	CompoNet Time Slot	ControlNet CTDMA	CAN CSMA/NBA	
1 Physical Layer	Ethernet	CompoNet	ControlNet	DeviceNet	

11.3.3 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 2: Level 1 + I/O Messages Server |
| Level 1 | Level 1: Explicit Messages Server |
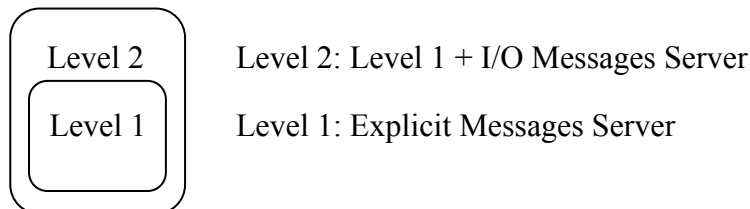
• Unconnected Message Manager (UCMM) client and server

• 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)

11.3.4 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.

11.3.5   Object Model

11.3.5.1   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.

11.3.5.2 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 150: 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 151: 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 Point Extended 1
6C hex	Analog Output Point Extended 1
6D hex	Discrete Input Point Extended 2
6E hex	Discrete Output Point Extended 2
6F hex	Analog Input Point Extended 2
70 hex	Analog Output Point Extended 2
71 hex	Discrete Input Point Extended 3
72 hex	Discrete Output Point Extended 3
73 hex	Analog Input Point Extended 3
74 hex	Analog Output Point Extended 3
80 hex	Module Configuration
81 hex	Module Configuration Extended 1

11.3.5.3 Explanation of the Table Headings in the Object Descriptions

Table 152: 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

11.3.5.4 Identity (01 hex)

The "Identity" class provides general information about the fieldbus coupler/controller that clearly identifies it.

Instance 0 (Class Attributes)

Table 153: 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 154: 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 155: 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

11.3.5.5 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 156: 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 157: 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 158: 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

11.3.5.6 Assembly Object (04 hex)

By means of the assembly classes, even several diverse objects can be combined. These could be, for example, input and output data, status and control information or diagnostic information. WAGO uses the manufacturer-specific instances in order to provide these objects for you in various arrangements. This gives you an efficient way to exchange process data. The following is a description of the individual static assembly instances with their contents and arrangements.

Static Assembly Instances – Overview

Table 159: Static assembly instances – Overview

Instance	Description
Instance 101 (65 hex)	for analog and digital output data
Instance 102 (66 hex)	for digital output data
Instance 103 (67 hex)	for analog output data
Instance 104 (68 hex)	for analog and digital input data and status
Instance 105 (69 hex)	for digital input data and status
Instance 106 (6A hex)	for analog input data and status
Instance 107 (6B hex)	for analog and digital input data
Instance 108 (6C hex)	for digital input data
Instance 109 (6D hex)	for analog input data

Instance 0 (Class Attributes)

Table 160: Assembly (04 hex) – Class

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

Instance 101 (65hex)

This assembly instance contains analog and digital output data.

Table 161: Static assembly instances – Instance 101 (65 hex)

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

Instance 102 (66_hex)

This assembly instance contains digital output data only.

Table 162: Static assembly instances – Instance 102 (66 _hex)

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

Instance 103 (67_hex)

This assembly instance contains analog output data only.

Table 163: Static assembly instances – Instance 103 (67 _hex)

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

Instance 104 (68_hex)

This assembly instance contains analog and digital input data and the status only.

Table 164: Static assembly instances – Instance 104 (68 _hex)

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

Instance 105 (69_hex)

This assembly instance contains digital input data and the status only.

Table 165: Static assembly instances – Instance 105 (69 _hex)

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

Instance 106 (6A_hex)

This assembly instance contains analog input data and the status only.

Table 166: Static assembly instances – Instance 106 (6A _hex)

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

Instance 107 (6B~hex~)

This assembly instance contains analog and digital input data.

Table 167: Static assembly instances – Instance 107 (6B $_{hex}$)

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

Instance 108 (6C~hex~)

This assembly instance contains digital input data.

Table 168: Static assembly instances – Instance 108 (6C $_{hex}$)

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

Instance 109 (6D~hex~)

This assembly instance contains analog input data.

Table 169: Static assembly instances – Instance 109 (6C $_{hex}$)

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

Instance 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 170: 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.

11.3.5.7 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.

11.3.5.8 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.

11.3.5.9 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 171: 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 172: 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 173: 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

11.3.5.10　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 174: 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 175: 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 176: 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

11.3.5.11 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 177: 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 178: 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 178: 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 179: 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 179: 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 180: 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 181: 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.

11.3.5.12 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 182: 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 183: 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

| 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 184: 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

11.3.5.13 Discrete Input Point (65 hex)

This class allows the reading of data of a particular digital input point.

Instance 0 (Class-Attributes)

Table 185: 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 186: 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 187: 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

11.3.5.14 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 188: 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 189: 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 190: 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

11.3.5.15 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 191: 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 192: 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 193: 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

11.3.5.16 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 194: 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 195: 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 196: 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

11.3.5.17 Discrete Output Point (66 hex)

This class enables data exchange for a particular digital output point.

Instance 0 (Class Attributes)

Table 197: 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 198: 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 199: 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

11.3.5.18 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 200: 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 201: 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 202: 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

11.3.5.19 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 203: 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 204: 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 205: 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

11.3.5.20 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 206: 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 207: 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 208: 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

11.3.5.21 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 209: 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 210: 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 211: 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

11.3.5.22 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 212: 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 213: 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 214: 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

11.3.5.23 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 215: 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 216: 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 217: 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

11.3.5.24 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 218: 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 219: 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 220: 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

11.3.5.25 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 221: 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 222: 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 223: 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

11.3.5.26 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 224: 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 225: 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 226: 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

11.3.5.27 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 227: 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 228: 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 229: 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

11.3.5.28 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 230: 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 231: 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 232: 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

11.3.5.29 Module Configuration (80 hex)

Instance 0 (Class Attributes)

Table 233: 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 234: 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 235: 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

11.3.5.30 Module Configuration Extended (81 hex)

The same as "Module Configuration (80 hex)" but with a description of module 255.

Instance 0 (Class Attributes)

Table 236: 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 237: 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 238: 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   I/O Modules

12.1   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, please
refer to the manuals for the I/O modules.

You will find these manuals on DVD ROM "AUTOMATION Tools and Docs"
(Item-no.: 0888-0412) or on the WAGO web pages under 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

12.2   Process Data Architecture for MODBUS/TCP

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

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

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

12.2.1 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.

12.2.1.1 1 Channel Digital Input Module with Diagnostics

750-435

Table 239: 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

12.2.1.2 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 240: 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

12.2.1.3 2 Channel Digital Input Module with Diagnostics

750-419, -421, -424, -425,
753-421, -424, -425

Table 241: 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

12.2.1.4 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 242: 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

12.2.1.5 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 243: 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

12.2.1.6 8 Channel Digital Input Modules

750-430, -431, -436, -437, -1415, -1416, -1417
753-430, -431, -434

Table 244: 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

12.2.1.7 16 Channel Digital Input Modules

750-1400, -1402, -1405, -1406, -1407

Table 245: 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

12.2.2   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.

12.2.2.1   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 246: 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

12.2.2.2   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 247: 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

12.2.2.3   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 248: 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 249: 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

12.2.2.4   4 Channel Digital Output Modules

750-504, -516, -519, -531,
753-504, -516, -531, -540

Table 250: 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

12.2.2.5   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 251: 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

12.2.2.6   8 Channel Digital Output Module

750-530, -536, -1515, -1516
753-530, -534

Table 252: 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

12.2.2.7   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 253: 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

12.2.2.8   16 Channel Digital Output Modules

750-1500, -1501, -1504, -1505

Table 254: 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 19	controls DO 8 Channel 18	controls DO 7 Channel 17	controls DO 6 Channel 16	controls DO 5 Channel 15	controls DO 4 Channel 14	controls DO 3 Channel 13	controls DO 2 Channel 12	controls DO 1 Channel 11

12.2.2.9   8 Channel Digital Input/Output Modules

750-1502, -1506

Table 255: 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

12.2.3   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.

> **i**   **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.

12.2.3.1   1 Channel Analog Input Modules

750-491, (and all variations)

Table 256: 1 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Measured Value U_D
1	D3	D2	Measured Value U_{ref}

12.2.3.2 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 257: 2 Channel Analog Input Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Measured Value Channel 1
1	D3	D2	Measured Value Channel 2

12.2.3.3 4 Channel Analog Input Modules

750-453, -455, -457, -459, -460, -468, (and all variations),
753-453, -455, -457, -459

Table 258: 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

12.2.4   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.

12.2.4.1   2 Channel Analog Output Modules

750-550, -552, -554, -556, -560, -562, 563, -585, (and all variations),
753-550, -552, -554, -556

Table 259: 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

12.2.4.2   4 Channel Analog Output Modules

750-553, -555, -557, -559,
753-553, -555, -557, -559

Table 260: 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

12.2.5   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.

12.2.5.1 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 261: 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 262: 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 263: 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

12.2.5.2 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 264: 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

12.2.5.3 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 265: 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	

12.2.5.4 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 266: 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		

12.2.5.5 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 267: Data Exchange Module

Input and Output Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Data bytes
1	D3	D2	

12.2.5.6 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 268: SSI Transmitter Interface Modules

Input Process Image			
Offset	Byte Destination		Description
	High Byte	Low Byte	
0	D1	D0	Data bytes
1	D3	D2	

12.2.5.7 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 269: Incremental Encoder Interface Modules 750-631/000-004, --010, -011

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	

Input 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	

Output Process Image

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 270: 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 271: Incremental Encoder Interface Modules 750-637

	Byte Destination		
Input and Output Process Image			
Offset	**High Byte**	**Low Byte**	**Description**
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 272: Digital Pulse Interface Modules 750-635

	Byte Destination			
Input and Output Process Image				
Offset	**High Byte**	**Low Byte**	**Description**	
0	D0	C0/S0	Data byte	Control/status byte
1	D2	D1	Data bytes	

12.2.5.8   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 273: DC-Drive Controller 750-636

	Byte Destination		
Input Process Image			
Offset	**High Byte**	**Low Byte**	**Description**

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

12.2.5.9   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 274: Steppercontroller 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)

12.2.5.10 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 275: 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		

12.2.5.11 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 276: DALI/DSI Master module 750-641

Input Process Image				
Offset	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
0	D0	S	DALI Response	Status byte
1	D2	D1	Message 3	DALI Address
2	D4	D3	Message 1	Message 2

Output Process Image

Offset	Byte Destination		Description	
	High Byte	**Low Byte**		
0	D0	C	DALI command, DSI dimming value	Control byte
1	D2	D1	Parameter 2	DALI Address
2	D4	D3	Command extension	Parameter 1

12.2.5.12 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 277: EnOcean Radio Receiver 750-642

Input Process Image

Offset	Byte Destination		Description	
	High Byte	**Low Byte**		
0	D0	S	Data byte	Status byte
1	D2	D1	Data bytes	

Output Process Image

Offset	Byte Destination		Description	
	High Byte	**Low Byte**		
0	-	C	not used	Control byte
1	-	-	not used	

12.2.5.13 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 278: MP Bus Master Module 750-643

Input and Output Process Image

Offset	Byte Destination		Description	
	High Byte	**Low Byte**		
0	C1/S1	C0/S0	extended Control/ Status byte	Control/status byte
1	D1	D0	Data bytes	
2	D3	D2		
3	D5	D4		

12.2.5.14 *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 279: 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		

12.2.5.15 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 280: Vibration Velocity/Bearing Condition Monitoring VIB I/O 750-645

Input and Output Process Image				
Offset	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
0	-	C0/S0	not used	Control/status byte (log. Channel 1, Sensor input 1)
1	D1	D0	Data bytes (log. Channel 1, Sensor input 1)	
2	-	C1/S1	not used	Control/status byte (log. Channel 2, Sensor input 2)
3	D3	D2	Data bytes (log. Channel 2, Sensor input 2)	
4	-	C2/S2	not used	Control/status byte (log. Channel 3, Sensor input 1)
5	D5	D4	Data bytes (log. Channel 3, Sensor input 3)	
6	-	C3/S3	not used	Control/status byte (log. Channel 4, Sensor input 2)
7	D7	D6	Data bytes (log. Channel 4, Sensor input 2)	

12.2.5.16 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 281: AS-interface Master module 750-655

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		

Input and Output Process Image

12.2.6   System Modules

12.2.6.1   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 282: 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

12.2.6.2   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 283: 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

12.3   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 coupler with EtherNet/IP.

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.

12.3.1   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).

12.3.1.1   1 Channel Digital Input Module with Diagnostics

750-435

Table 284: 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).

12.3.1.2   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 285: 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).

12.3.1.3   2 Channel Digital Input Module with Diagnostics

750-419, -421, -424, -425,
753-421, -424, -425

Table 286: 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).

12.3.1.4   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 287: 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).

12.3.1.5   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 288: 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).

12.3.1.6   8 Channel Digital Input Modules

750-430, -431, -436, -437, -1415, -1416, -1417
753-430, -431, -434

Table 289: 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).

12.3.1.7   16 Channel Digital Input Modules

750-1400, -1402, -1405, -1406, -1407

Table 290: 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

12.3.2   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).

12.3.2.1   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 291: 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).

12.3.2.2   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 292: 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).

12.3.2.3   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 293: 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 294: 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).

12.3.2.4   4 Channel Digital Output Modules

750-504, -516, -519, -531,
753-504, -516, -531, -540

Table 295: 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).

12.3.2.5   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 296: 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).

12.3.2.6   8 Channel Digital Output Module

750-530, -536, -1515, -1516
753-530, -534

Table 297: 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).

12.3.2.7   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 298: 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).

12.3.2.8   16 Channel Digital Output Modules

750-1500, -1501, -1504, -1505

Table 299: 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).

12.3.2.9   8 Channel Digital Input/Output Modules

750-1502, -1506

Table 300: 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 modules represent 2x2 bytes and seize 2 Instances in Class (0x67).

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 modules represent 2x2 bytes and seize 2 Instances in Class (0x67).

12.3.3   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 .

12.3.3.1   1 Channel Analog Input Modules

750-491, (and all variations)

Table 301: 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).

12.3.3.2   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 302: 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).

12.3.3.3   4 Channel Analog Input Modules

750-453, -455, -457, -459, -460, -468, (and all variations),
753-453, -455, -457, -459

Table 303: 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).

12.3.4   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.

12.3.4.1   2 Channel Analog Output Modules

750-550, -552, -554, -556, -560, -562, 563, -585, (and all variations),
753-550, -552, -554, -556

Table 304: 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).

12.3.4.2   4 Channel Analog Output Modules

750-553, -555, -557, -559,
753-553, -555, -557, -559

Table 305: 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).

12.3.5   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).

12.3.5.1   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 306: 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 307: 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 308: Counter Modules 750-638, 753-638

| Instance | Byte Destination | | Description |
	High Byte	Low Byte	
Input Process Image			
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).

| Instance | Byte Destination | | Description |
	High Byte	Low Byte	
Output Process Image			
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).

12.3.5.2   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 309: Pulse Width Modules 750-511, /xxx-xxx

| Instance | Byte Destination | | Description |
	High Byte	Low Byte	
Input and Output Process			
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).

12.3.5.3   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 310: 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).

12.3.5.4 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 311: 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).

12.3.5.5   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 312: 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).

12.3.5.6   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 313: 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 314: SSI Transmitter Interface I/O Modules with an Alternative Data Format

Input Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**High Byte**		
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).

12.3.5.7 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 315: Incremental Encoder Interface Modules 750-631/000-004, -010, -011

Input Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
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**		**Description**
	High Byte	**Low Byte**	
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 316: 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 317: 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 318: Incremental Encoder Interface Modules 750-635, 750-635

| 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).

12.3.5.8 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 319: DC-Drive Controller 750-636

| 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).

12.3.5.9   **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 320: 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).

12.3.5.10 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 321: 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).

12.3.5.11 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 322: 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).

12.3.5.12 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 323: 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).

12.3.5.13 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 324: MP Bus Master Module 750-643

Input and Output Process Image				
Instance	**Byte Destination**		**Description**	
	High Byte	**Low Byte**		
n	C1/S1	C0/S0	extended Control/ Status byte	Control/status byte
	D1	D0	Data bytes	
	D3	D2		
	D5	D4		

The specialty modules represent 1x8 bytes input and output data and seize 1 Instance in Class (0x67) and 1 Instance in Class (0x68).

12.3.5.14 *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 325: 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.

12.3.5.15 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 326: 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).

12.3.5.16 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 327: 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).

12.3.6   System Modules

12.3.6.1   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 328: 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).

12.3.6.2   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 329: 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).

13   Application Examples

13.1   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

13.2   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 330: 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".
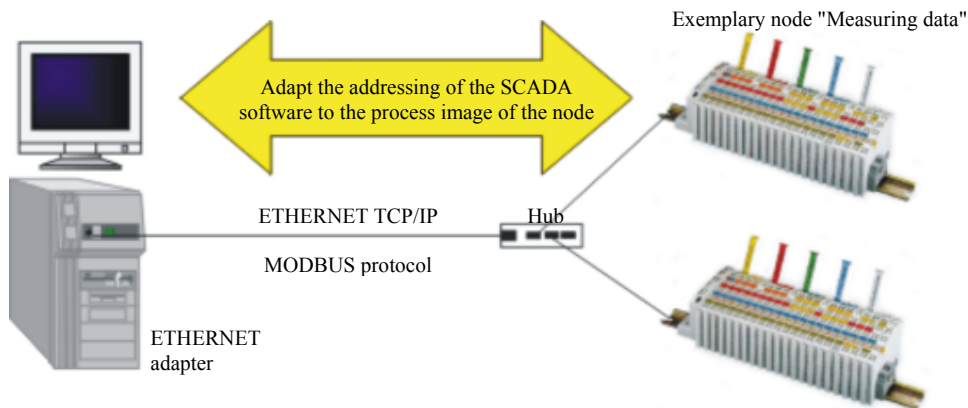
Figure 60: Example SCADA software with MODBUS driver

Information

Additional Information
Please refer to the respective SCADA product manual for a detailed description of the particular software operation.

14   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.

14.1   Identification

14.1.1   For Europe according to CENELEC and IEC
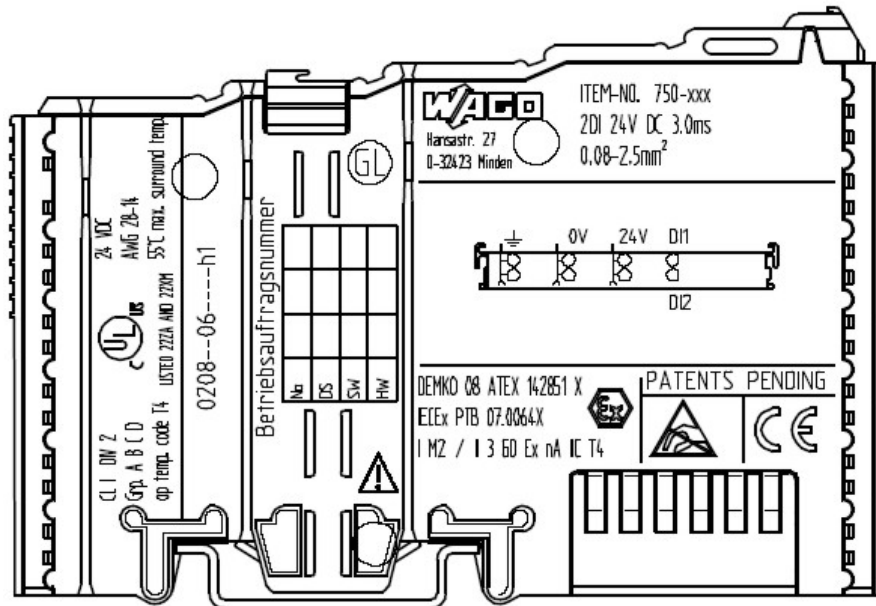
Figure 61: Example for lateral labeling of bus modules

Figure 62: Printing on text detail in accordance with CENELEC and IEC

Table 331: Description of Printing on

Printing on Text	Description
DEMKO 08 ATEX 142851 X IECEx PTB 07.0064X	Approval body and/or number of the examination certificate
I M2 / II 3 GD	Explosion protection group and Unit category
Ex nA	Type of ignition and extended identification
IIC	Explosion protection group
T4	Temperature class
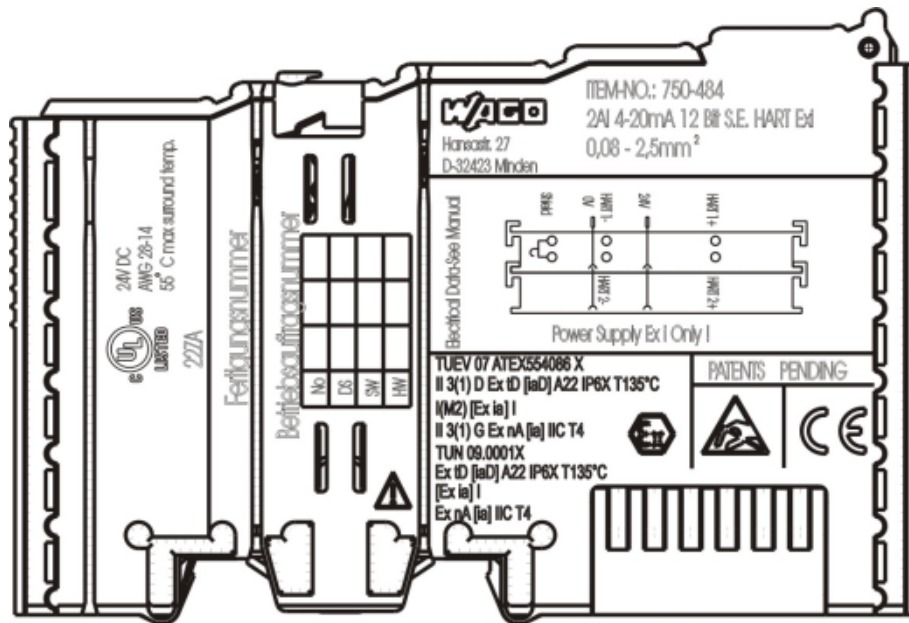
Figure 63: Example of side marking of Ex i and IEC Ex i approved I/O modules

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 64: Inscription text detail acc. CENELEC and IEC

Table 332: Description of the inscription

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

14.1.2   For America according to NEC 500
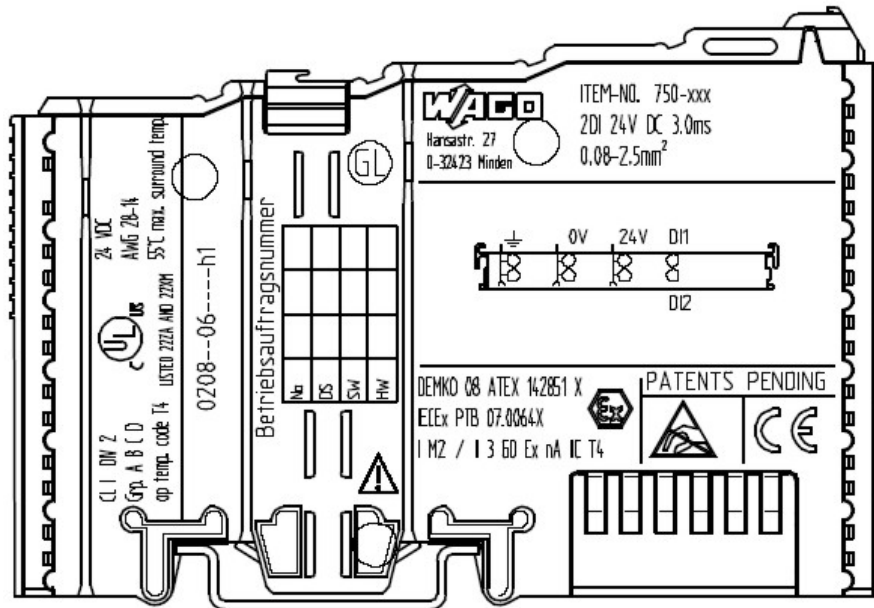
Figure 65: Example for lateral labeling of bus modules

Figure 66: Printing on text detail in accordance with NEC

Table 333: Description of Printing on

Printing on Text	Description
CL 1	Explosion protection group (condition of use category)
DIV 2	Area of application (zone)
Grp. ABCD	Explosion group (gas group)
Optemp code T4	Temperature class

14.2   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 334: 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 335: 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

<div style="border:1px solid">

NOTICE

Notice the following points
When using the **WAGO-I/O SYSTEM 750** (electrical operation) with Ex approval, the following points are mandatory:

</div>

14.2.1   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-SYSTEMs 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.

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 -20 °C to +55 °C.

14.2.2   Special Conditions for Safe Ooperation of the Ex i (acc. TÜV 07 ATEX 554086 X)

1.   For operation as a Category 3 Device (in Zone 2 or 22), the WAGO-I/O-SYSTEM 750-*** must be mounted in an enclosure that fulfills the requirements of the directive 94/9/EG and the relevant standards (see designation) EN 60079-0, EN 60079-11, EN 60079-15, EN 61241-0 and EN 61241-1. For operation as a Group I Category M2 device, the device must be mounted in a housing that ensures adequate protection in accordance with EN 61241-0 and EN 61241-1. A declaration of conformity according to Appendix X of Directive 94/9/EG must confirm the correct installation of the devices above in the enclosure or switchgear cabinet.

2.   If the interface circuits are operated without the fieldbus coupler station of type 750-3../...-... (DEMKO 08 ATEX 142851 X), then measures must be taken outside of the device so that the rated voltage will not be exceeded by more than 40% due to temporary faults.

3.   DIP switches, coding switches and potentiometers that are connected to the module may only be operated if an explosive atmosphere can be ruled out.

4.   Non-intrinsically safe circuits may only be connected and disconnected for installation, maintenance and repair. Explosive atmosphere and installation, maintenance or repair occurring simultaneously must be ruled out.

5.   For types 750-606, 750-625/000-001, 750-487/003-000, 750-484, the following must be taken into account: The interface circuits must be limited to overvoltage category I/II/III (electrical circuits without power supply/electrical circuits with power supply) as defined in EN 60664-1.

6.   The permissible ambient temperature range is -20 °C to +55 °C.

14.2.3   Special Conditions for the Safe Operation of the IEC Ex i (acc. TUN 09.0001 X)

1.   For operation as a Dc or Gc device (in Zone 2 or 22), the WAGO-I/O-SYSTEM 750-*** must be mounted in an enclosure that fulfills the requirements for a device of the relevant standards (see designation) IEC 60079-0, IEC 60079-11, IEC 60079-15, IEC 61241-0 and IEC 61241-1. For operation as a Group I Category M2 device, the device must be mounted in a housing that ensures adequate protection in accordance with IEC 61241-0 and IEC 61241-1. A declaration of conformity must confirm compliance with these requirements and correct installation of the devices above in the enclosure or switchgear cabinet by an Ex certification authority.

2.   Outside the device, measures must be taken so that the rated voltage will not be exceeded by more than 40% due to temporary faults.

3.   DIP switches, coding switches and potentiometers that are connected to the module may only be operated if an explosive atmosphere can be ruled out.

4.   Non-intrinsically safe circuits may only be connected and disconnected for installation, maintenance and repair. Explosive atmosphere and installation, maintenance or repair occurring simultaneously must be ruled out.

5.   For types 750-606, 750-625/000-001, 750-487/003-000, 750-484, the following must be taken into account: The interface circuits must be limited to overvoltage category I/II/III (electrical circuits without power supply/electrical circuits with power supply) as defined in EN 60664-1.

6.   The permissible ambient temperature range is -20 °C to +55 °C.

14.2.4   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.

NOTICE

Explosion hazard!
Explosion hazard - substitution of components may impair suitability for Class I, Div. 2.

NOTICE

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.

When a fuse is provided, the following marking shall be provided:
"A switch suitable for the location where the equipment is installed shall be provided to remove the power from the fuse".
The switch need not be integrated in the equipment.

For devices with Ethernet connectors:
"Only for use in LAN, not for connection to telecommunication circuits".

NOTICE

Use only with antenna module 758-910!
Use Module 750-642 only with antenna module 758-910.

Information

i

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.

List of Figures

List of Tables