# WAGO-I/O-SYSTEM 750

## Manual



# 750-806
## Fieldbus Controller DeviceNet™
## 16-bit CPU

Version 2.0.0

**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

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

## 1.1    Validity of this Documentation

This documentation is only applicable to the "Fieldbus Controller DeviceNet™" (750-806).

The product "Fieldbus Controller DeviceNet™" (750-806) 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'<br>'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.<br>e.g.: *C:\Program Files\WAGO Software* |
| **Menu** | Menu items are marked in bold letters.<br>e.g.: **Save** |
| > | A greater-than sign between two names means the selection of a menu item from a menu.<br>e.g.: **File** > **New** |
| **Input** | Designation of input or optional fields are marked in bold letters,<br>e.g.: **Start of measurement range** |
| "Value" | Input or selective values are marked in inverted commas.<br>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.<br>e.g.: **[Input]** |
| **[Key]** | Keys are marked with bold letters in square brackets.<br>e.g.: **[F5]** |

# 2      Important Notes

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

## 2.1     Legal Bases

### 2.1.1    Subject to Changes

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

### 2.1.2    Personnel Qualifications

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

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

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

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

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

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

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

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

## 2.2    Safety Advice (Precautions)

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

⚠ **DANGER**

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

⚠ **DANGER**

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

**NOTICE**

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

**NOTICE**

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

**NOTICE**

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

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

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

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

# 3      System Description

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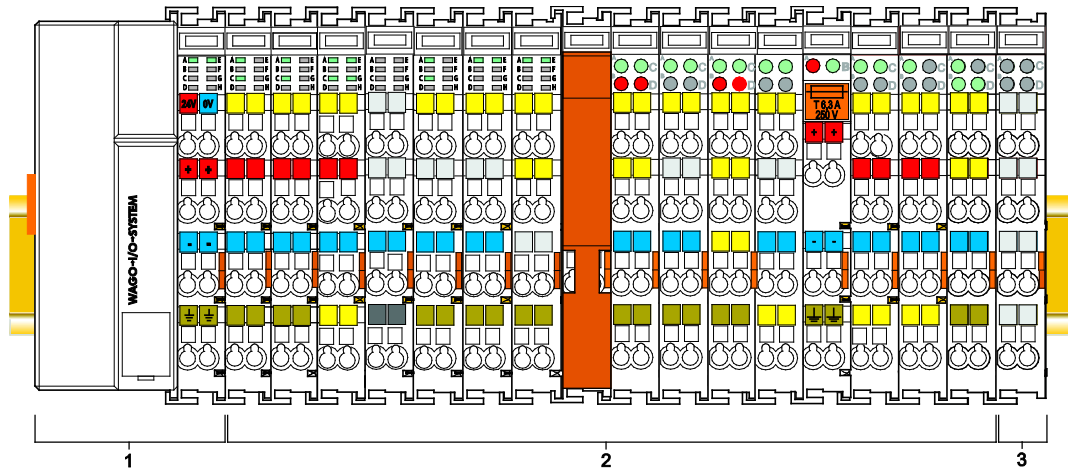


Figure 1: Fieldbus Node (Example)

Fieldbus couplers/controllers are available for different fieldbus systems.

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

The fieldbus coupler/controller exchanges process data with the respective control
via the respective fieldbus. The programmable fieldbus controllers (PFC) allow
implementation of additional PLC functions. WAGO-I/O-*PRO* is used to program
the fieldbus controllers according to IEC 61131-3.

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

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

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

## 3.1     Manufacturing Number

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

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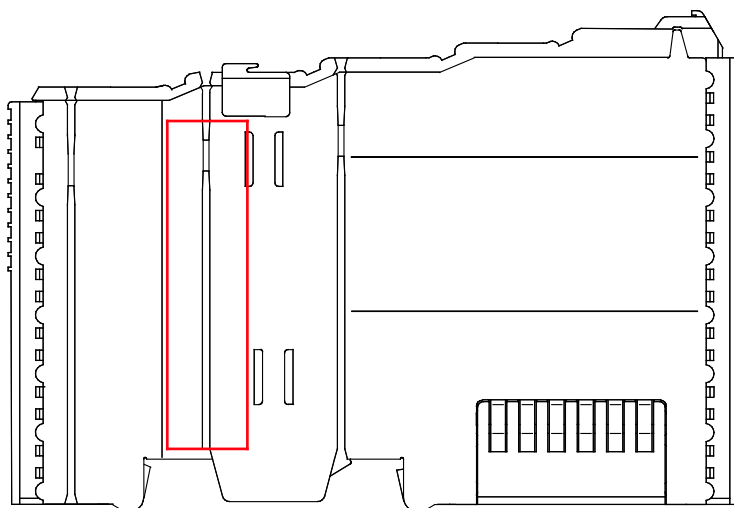


Figure 2: Marking Area for Serial Numbers

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

Example structure of the rows: 0114010101…

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

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

## 3.2    Component Update

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

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

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

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

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

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

## 3.3    Storage, Assembly and Transport

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

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

## 3.4 Assembly Guidelines/Standards

- DIN 60204      Electrical equipping of machines

- DIN EN 50178   Equipping of high-voltage systems with electronic
  components    (replacement for VDE 0160)

- EN 60439      Low-voltage switchgear and controlgear assemblies

# 3.5      Power Supply

## 3.5.1      Isolation

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

•      Electrically isolated fieldbus interface via transformer

•      Electronics of the fieldbus couplers/controllers and the I/O modules
       (internal bus)

•      All I/O modules have an electrical isolation between the electronics
       (internal bus, logic) and the field electronics. Some digital and analog input
       modules have each channel electrically isolated, please see catalog.
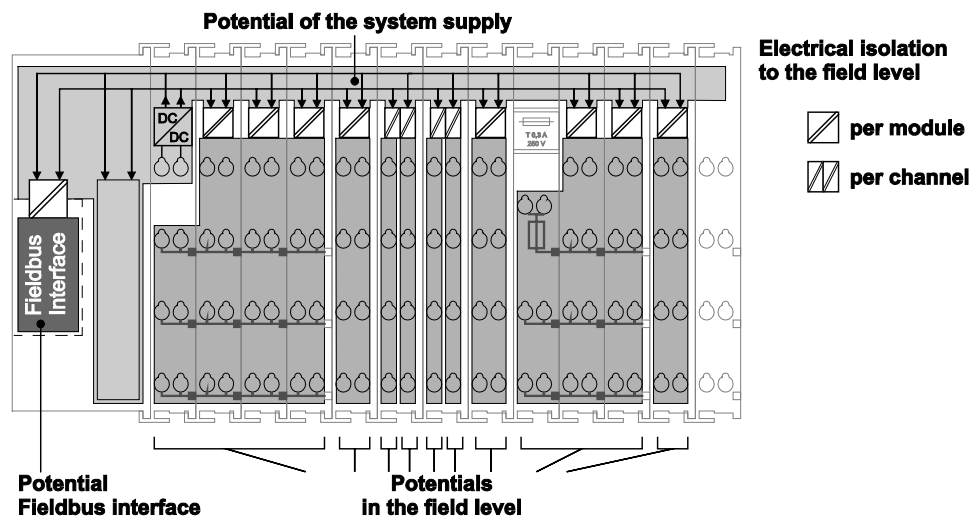


Figure 3: Isolation for Fieldbus Couplers/Controllers (Example)

## 3.5.2    System Supply

### 3.5.2.1    Connection

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

**NOTICE**

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



Figure 4: System Supply via Fieldbus Coupler/Controller (left) and via Internal System Supply Module (right)

Table 3: Legend for Figure "System Supply via Fieldbus Coupler/Controller (left) and via Internal System Supply Module (right)"

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

The fed DC 24 V supplies all internal system components, e.g. fieldbus coupler/controller electronics, fieldbus interface and I/O modules via the internal bus (5 V system voltage). The 5 V system voltage is galvanically connected to the 24 V system supply.
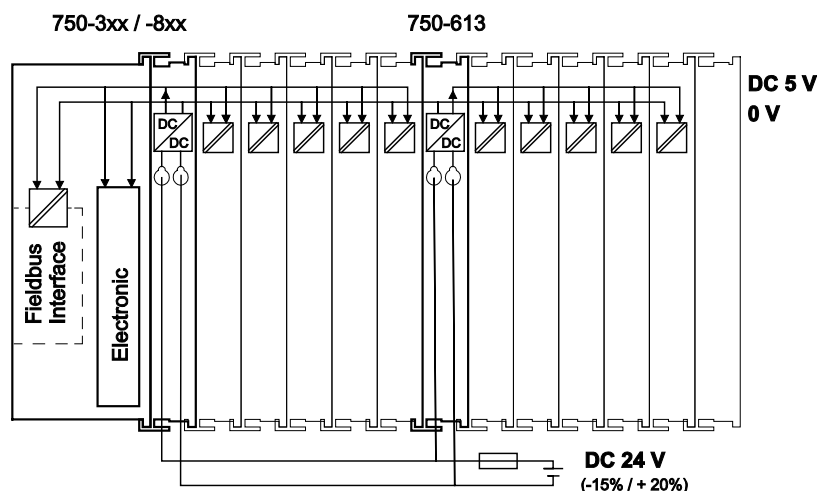
Figure 5: System Voltage for Standard Couplers/Controllers and Extended ECO Couplers

---

→ **Note**

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

---

### 3.5.2.2    Dimensioning

---

→ **Note**

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

---

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

Table 4: Alignment

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

[*]  See current catalog, manuals, Internet

**Example:**

**Calculating the current consumption on the fieldbus coupler:**

| | |
|---|---|
| Internal current consumption of the coupler | 350 mA at 5 V |
| Total current for I/O modules | 1650 mA at 5 V |
| **Sum I$_{(5\ V)\ total}$** | **2000 mA at 5 V** |

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

---

### Note

**Please note the aggregate current for I/O modules. It may be necessary to supply potential!**
When the sum of the internal current consumption for the I/O modules exceeds their aggregate current, you must use a supply module with bus power supply. Install it before the position where the permissible aggregate current would be exceeded.

---

**Example:**

**Calculating the total current on a standard fieldbus coupler/controller:**

A node configuration with 20 relay modules (750-517) and 30 digital input modules (750-405) should be attached to a fieldbus coupler/controller:

| | |
|---|---|
| Internal current consumptions | $20 \times 90$ mA = 1800 mA at 5 V |
| | $+\ 30 \times\ 2$ mA =   60 mA at 5 V |
| **Sum of internal current consumptions** | **1860 mA at 5 V** |

However, the fieldbus coupler can only provide 1650 mA for the I/O modules. Consequently, an internal system supply module (750-613), e. g. in the middle of the node, should be added.

---

### Note

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

---

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

**Fieldbus coupler or controller**

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

**Internal system supply module**

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

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

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

---

## Note

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

---

## 3.5.3    Field Supply

### 3.5.3.1    Connection

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

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

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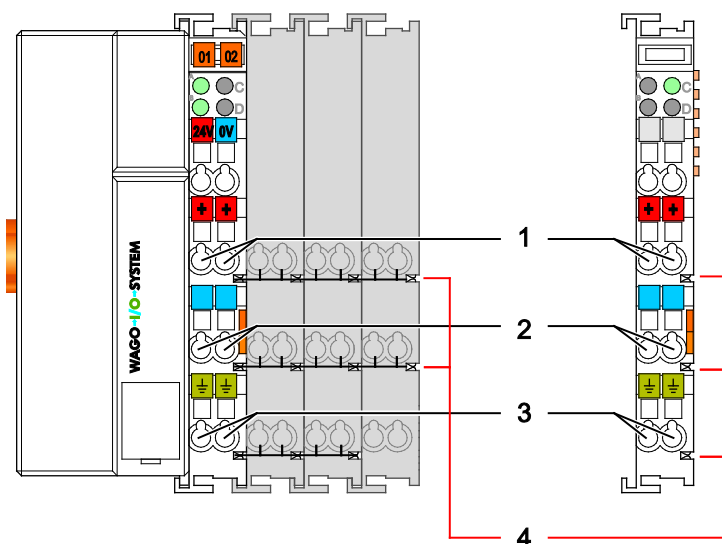


Figure 6: Field Supply for Standard Couplers/Controllers and Extended ECO Couplers

Table 5: Legend for Figure "Field Supply for Standard Couplers/Controllers and Extended ECO Couplers"

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

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

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

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

## Note

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

## Note

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

### 3.5.3.2   Fusing

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

Table 6: Power Supply Modules

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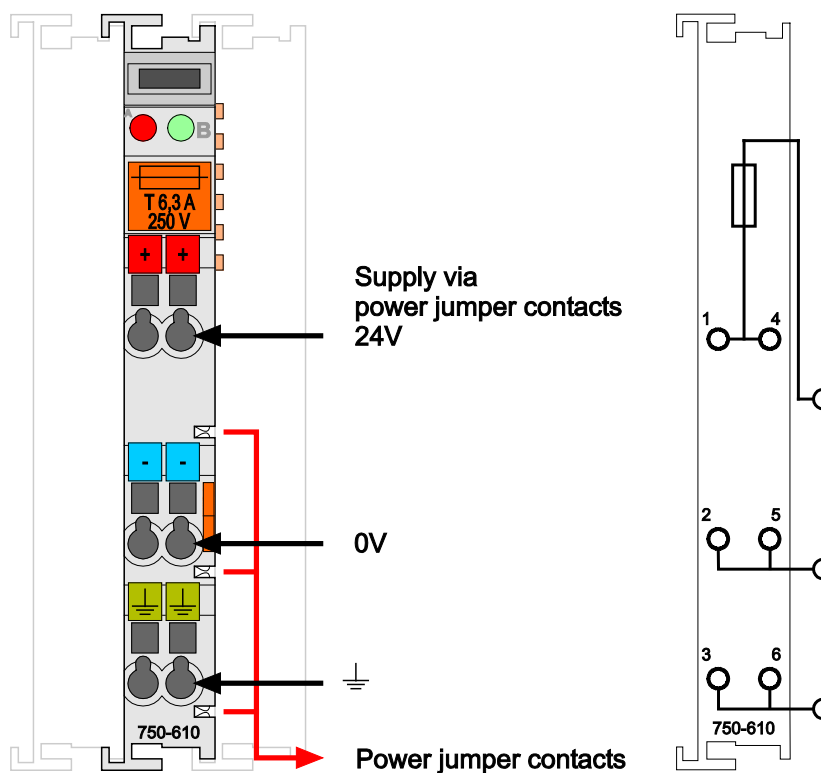


Figure 7: Supply Module with Fuse Carrier (Example 750-610)

**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 I/O
modules, the fuse holder may be pulled out. In order to do this, use a screwdriver

for example, to reach into one of the slits (one on both sides) and pull out the holder.



Figure 8: Removing the Fuse Carrier

Lifting the cover to the side opens the fuse carrier.



Figure 9: Opening the Fuse Carrier



Figure 10: Changing the Fuse

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

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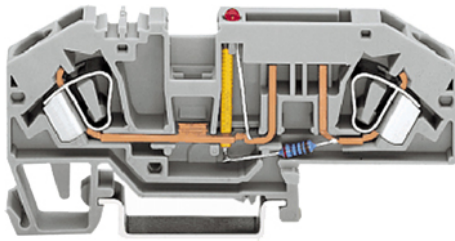


Figure 11: Fuse Modules for Automotive Fuses, Series 282
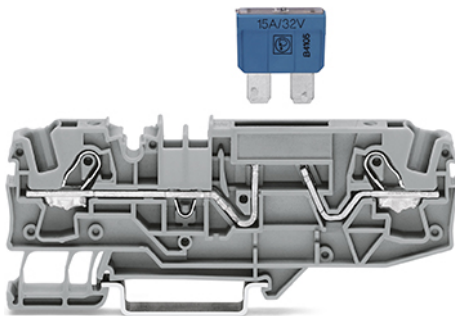


Figure 12: Fuse Modules for Automotive Fuses, Series 2006
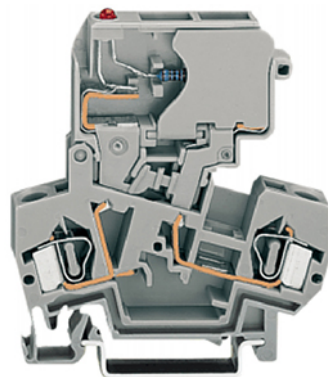


Figure 13: Fuse Modules with Pivotable Fuse Carrier, Series 281
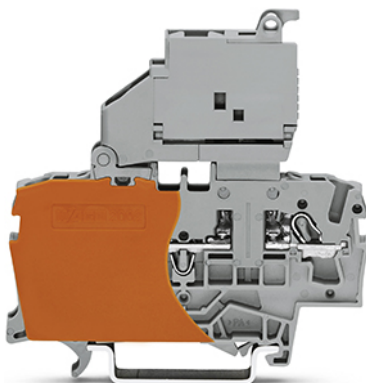


Figure 14: Fuse Modules with Pivotable Fuse Carrier, Series 2002

## 3.5.4    Supplementary Power Supply Regulations

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

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

Table 7: Filter Modules for 24 V Supply

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

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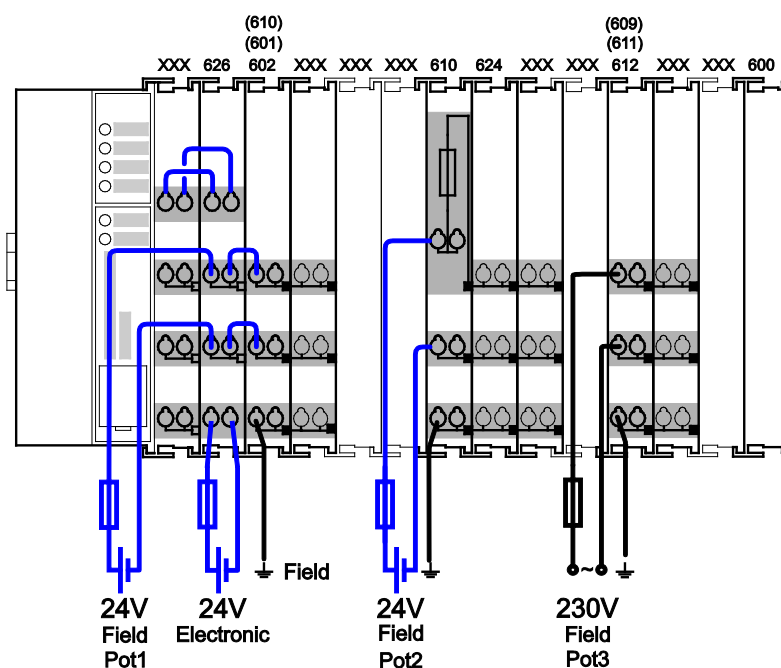


Figure 15: Power Supply Concept

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

## 3.5.5     Supply Example

> **Note**
>
> **The system supply and the field supply shall be separated!**
> You should separate the system supply and the field supply in order to ensure bus operation in the event of a short-circuit on the actuator side.
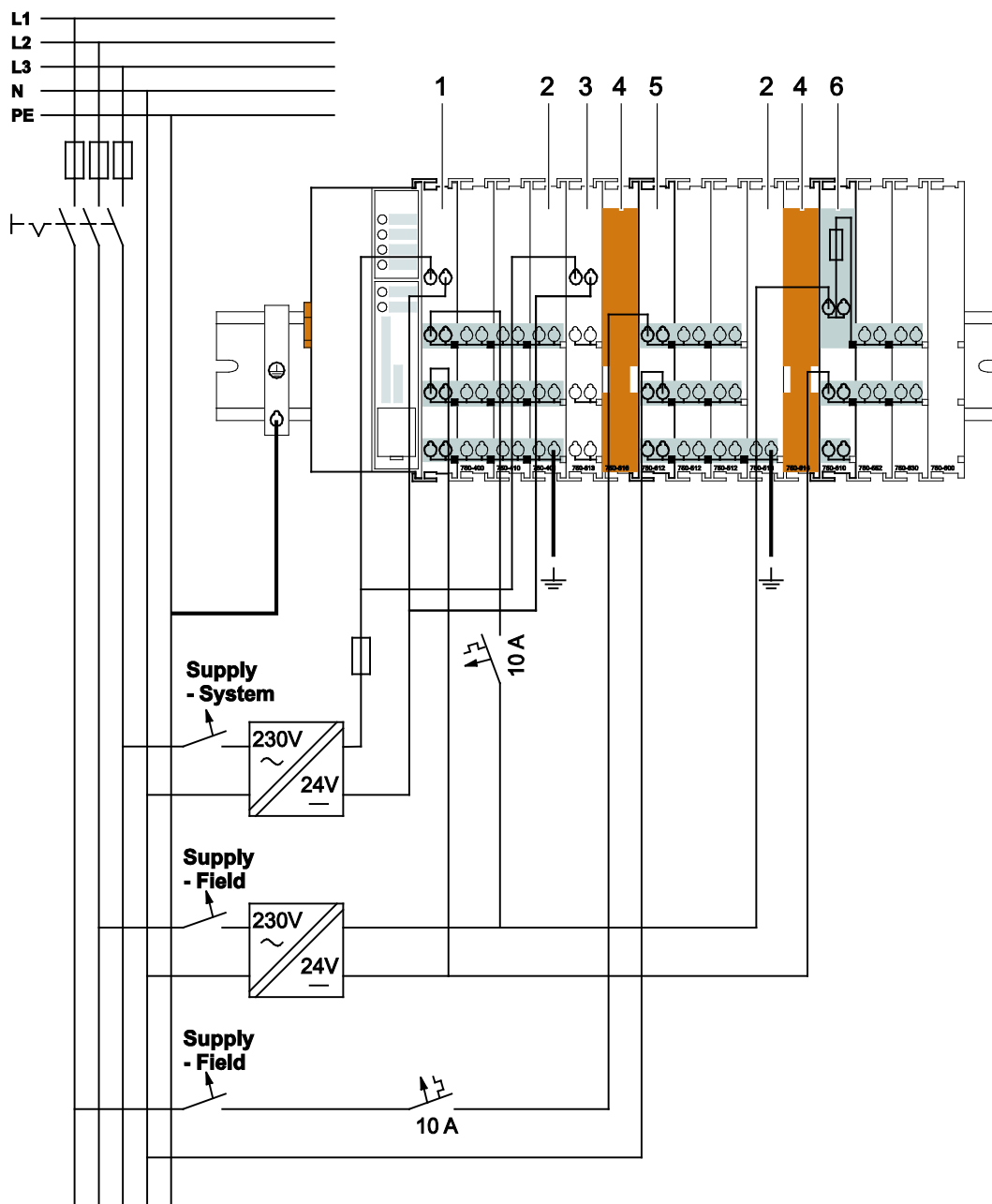
Figure 16: Supply Example for Standard Couplers/Controllers

Table 8: Legend for Figure "Supply Example for Fieldbus Coupler/Controller"

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

## 3.5.6    Power Supply Unit

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

---

→    **Note**

**Recommendation**
A stable power supply cannot always be assumed everywhere. Therefore, you
should use regulated power supplies to ensure the quality of the supply voltage
(see also table "WAGO power supply units").

---

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

---

→    **Note**

**Power failure time not acc. IEC 61131-2!**
Note that the power failure time of 10 ms acc. IEC 61131-2 is not maintained in a
maximum configuration.

---

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

---

→    **Note**

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

---

Table 9: WAGO Power Supply Units (Selection)

| 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 |
| | Rail-mounted modules with universal mounting carrier |
| 288-809 | AC 115 V/DC 24 V; 0,5 A |
| 288-810 | AC 230 V/DC 24 V; 0,5 A |
| 288-812 | AC 230 V/DC 24 V; 2 A |
| 288-813 | AC 115 V/DC 24 V; 2 A |

# 3.6 Grounding

## 3.6.1 Grounding the DIN Rail

### 3.6.1.1 Framework Assembly

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

## ⚠ DANGER

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

### 3.6.1.2 Insulated Assembly

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

## Note

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

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

Table 10: WAGO Ground Wire Terminals

| Order No. | Description |
|-----------|-------------|
| 283-609   | 1-conductor ground (earth) terminal block make an automatic contact to the carrier rail; conductor cross section: 0.2 mm² … 16 mm² **Note**: Also order the end and intermediate plate (283-320). |

## 3.6.2    Grounding Function

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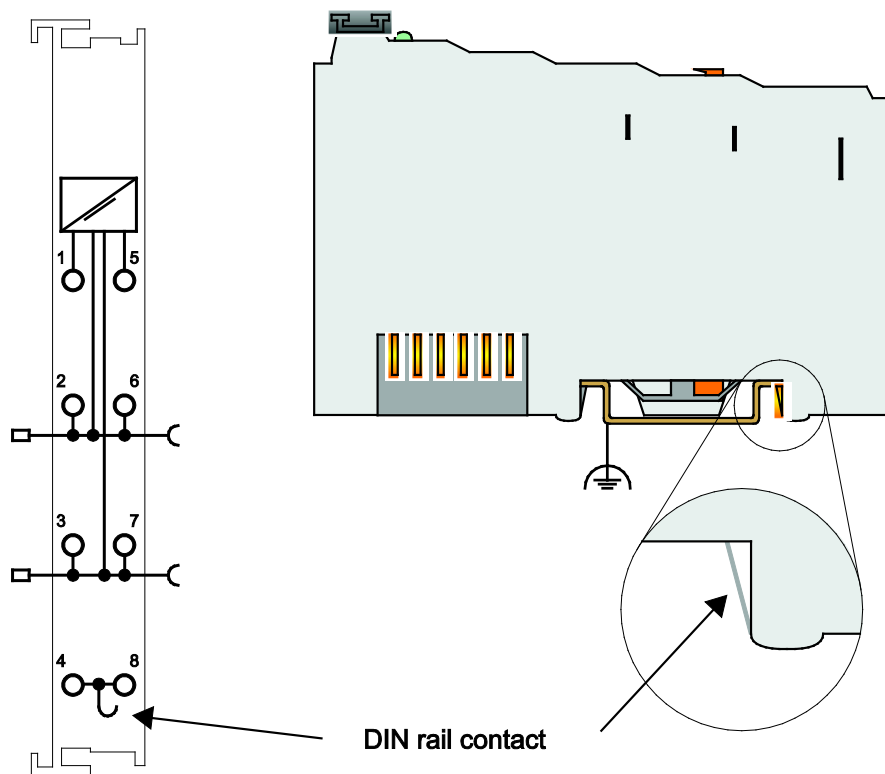


DIN rail contact

Figure 17: Carrier Rail Contact (Example)

⚠ **DANGER**

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

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

# 3.7    Shielding

## 3.7.1    General

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

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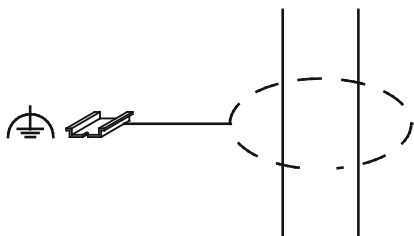


Figure 18: Cable Shield at Ground Potential

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

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

## 3.7.2    Bus Cables

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

### 3.7.3    Signal Lines

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

---

**Note**

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

---

### 3.7.4    WAGO Shield Connecting System

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


Figure 19: Examples of the WAGO Shield Connecting System


Figure 20: Application of the WAGO Shield Connecting System

# 4        Device Description

The Fieldbus Controller DeviceNet[TM] 750-806 links the
WAGO-I/O-SYSTEM 750 to the DeviceNet[TM] fieldbus system.

This fieldbus controller can be used for applications in mechanical and systems
engineering, as well as in the processing industry.

The fieldbus connection is made via series 231 5-pin plug connector
*MULTI CONNECTION SYSTEM (MCS)*.

The DIP switch can be used to specify baud rate and station address of the
fieldbus controller.

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

---

**Information**

**Compatibility with IEC-61131-3 programming software!**
The compatibility between your fieldbus controller and the IEC 61131-3
programming software used depends on the fieldbus controller's firmware
version and the version of the programming software.
A list of recommended combinations can be found at www.wago.com.
Use the search function (search for "Compatibility Notes").

---

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

The fieldbus controller has  128 KB program memory, 64 KB data memory and
8 KB retentive memory available for the IEC 61131-3 programming.

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

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

The bits of the digital modules are combined into bytes and then mapped after the
analog ones in the process image. If the number of digital I/Os is greater than 8
bits, the Fieldbus Controller automatically begins a new byte.

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

---

The fieldbus controller supports the following DeviceNet$^{TM}$ functions:

- Dynamic connection setup via the "Unconnected Message Manager Port" (UCMM) with up to 5 subscribers at the same time
- Addressing the node via "Offline Connection Set" (e.g., in case of error)
- Defined node unsubscribe while controlling via "DeviceNet Shutdown" (in case of error for safety-related applications)
- Compiling "Assembly Instances" during data transmission with process data from various application objects or via "Dynamic Assembly"
- Changing station address (called "MAC ID" with DeviceNet$^{TM}$) via "Change MAC ID by SW"
- Configurable communication monitoring via "Heartbeat"
- Synchronization of the slaves via "Bit-Strobe" multicast message from the master

# 4.1   View

The view below shows the three parts of the device:

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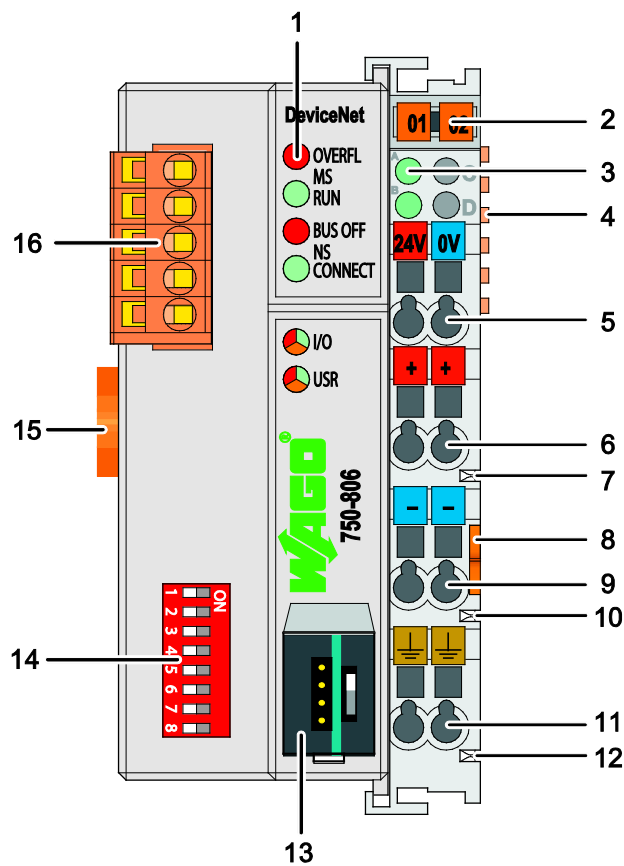


Figure 21: View DeviceNet™ Fieldbus Controller

Table 11: Legend for Figure "View DeviceNet<sup>TM</sup> Fieldbus Controller"

| Pos. | Desig-nation | Meaning | Details see Section |
|------|------|---------|---------------------|
| 1 | OVERFL, RUN, BUS OFF, CONNECT | Status LEDs Fieldbus | "Device Description" > "Display Elements" |
| 2 | --- | Group marking carrier (retractable) with additional marking possibility on two miniature WSB markers | --- |
| 3 | A, B or C | Status LED's System/Field Supply | "Device Description" > "Display Elements" |
| 4 | --- | Data Contacts | "Connect Devices" > "Data Contacts/Internal Bus" |
| 5 | 24 V, 0 V | CAGE CLAMP® Connections System Supply | "Connect Devices" > "Connecting a conductor to the CAGE CLAMP®" |
| 6 | + | CAGE CLAMP® Connections Field Supply 24 VDC | "Connect Devices" > "Connecting a conductor to the CAGE CLAMP®" |
| 7 | --- | Power Jumper Contact 24 VDC | "Connect Devices" > "Power Contacts/ Field Supply" |
| 8 | --- | Unlocking Lug | "Mounting" > "Inserting and Removing Devices" |
| 9 | - | CAGE CLAMP® Connections Field Supply 0 V | "Connect Devices" > "Connecting a conductor to the CAGE CLAMP®" |
| 10 | --- | Power Jumper Contact 0 V | "Connect Devices" > "Power Contacts/ Field Supply" |
| 11 | (Ground) | CAGE CLAMP® Connections Field Supply (Ground) | "Connect Devices" > "Connecting a conductor to the CAGE CLAMP®" |
| 12 | --- | Power Jumper Contact (Ground) | "Connect Devices" > "Power Contacts/ Field Supply" |
| 13 | --- | Service Interface (open flap) | "Device Description" > "Operating Elements" |
| 16 | --- | DIP Switch | "Device Description" > "Operating Elements" |
| 15 | --- | Locking Disc | "Mounting" > "Inserting and Removing Devices" |
| 16 | --- | Fieldbus Data contacts connection, 231 Series (MCS) | "Device Description" > "Connectors" |

# 4.2    Connectors

## 4.2.1    Device Supply

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

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

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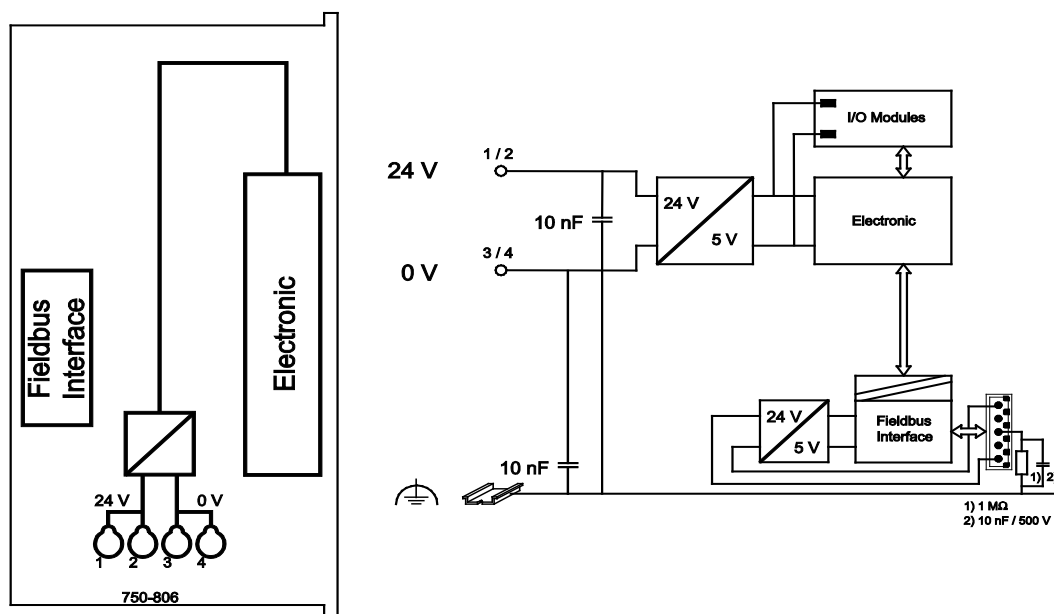


Figure 22: Device Supply

## 4.2.2    **Fieldbus Connection**

The fieldbus connection for DeviceNet<sup>TM</sup> is made via Series 231 5-pin plug connector from the *MULTI CONNECTION SYSTEM (MCS)*. A connector (OpenStyle) is the counterpart. The 231-305/010-000/050-000 connector is included.
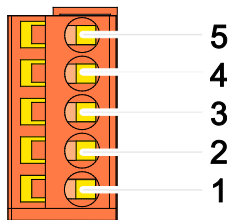


Figure 23: Fieldbus Connections, Series 231 *(MCS)*

Table 12: Pin Assignment for the Fieldbus Connection, Series 231 *(MCS)*

| PIN | Signal | Code$^*$ | Description |
|---|---|---|---|
| 5 | V+ | red | 11 V … 25 V |
| 4 | CAN_H | white | CAN Signal $_{High}$ |
| 3 | Drain shield | | Shield termination |
| 2 | CAN_L | blue | CAN Signal $_{Low}$ |
| 1 | V- | black | 0 V |

$^*$ according to DeviceNet<sup>TM</sup> specification, identical to the conductors of the DeviceNet<sup>TM</sup> cable

To connect small conductor cross-sections, an insulation stop of series 231-670 (white), 231-671 (light gray) or 231-672 (dark gray) should be used due to their flexibility. The insulating stop prevents the conductor from deforming when pushed against the conductor stop. As a result, the conductor insulation may be clamped, causing intermittent contact or no contact at all. The marking of the connector, as well as housing parts, test plugs with cable and male connectors for cable extensions are available.

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

DC/DC converters and optocouplers in the fieldbus interface electrically isolate the fieldbus system and the electronics.

# 4.3    Display Elements

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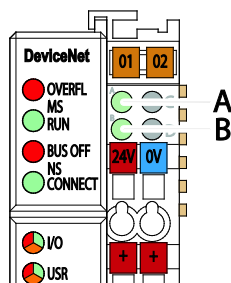


Figure 24: Display Elements

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

Table 13: Display Elements Fieldbus Status

| LED | Color | Meaning |
|---|---|---|
| OVERFL | red | Indicates an error or defect on the fieldbus controller |
| RUN | green | Indicates that the fieldbus controller is operational |
| BUS OFF | red | Indicates an error or malfunction in the network |
| CONNECT | green | Indicates that the fieldbus controller is ready for network communication |

Table 14: Display Elements Node Status

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

Table 15: Display Elements Supply Voltage

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

## Information

**More information about the LED Signaling**
Read the detailed description for the evaluation of the displayed LED state in the section "Diagnostics" > … > "LED Signaling".

# 4.4      Operating Elements

## 4.4.1     Service Interface

The service interface is located behind the flap.

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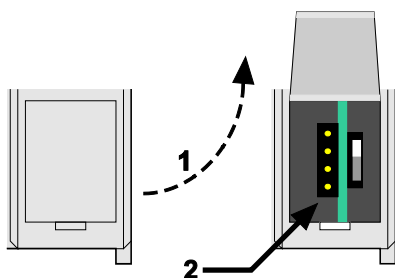


Figure 25: Service Interface (Closed and Opened Flap)

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

| Number | Description |
|--------|-------------|
| 1      | Open closed flap |
| 2      | View Service Interface |

**NOTICE**

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

The connection to the 4-pin header under the cover flap can be realized via the
communication cables with the item numbers750-920 and 750-923 or via the
WAGO radio adapter with the item number 750-921.

## 4.4.2   Mode Selector Switch

The mode selector switch is located behind the cover flap.
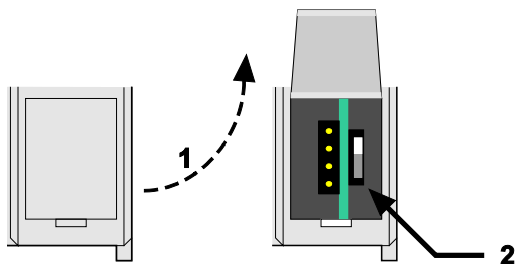


Figure 26: Mode Selector Switch (Closed and Open Damper of the Service Port)

Table 17: Legend for Figure „Mode Selector Switch"

| Number | Description |
|---|---|
| 1 | Open the damper |
| 2 | Mode selector switch |

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

**NOTICE**

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

**Note**

**Pre-programming the outputs for program stop!**
You have the ability to program the behavior of the fieldbus controller so that the
outputs switch in a safe condition in the case of program stop.
For this WAGO-I/O-*PRO* allocates a function with *GET_STOP_VALUE* (library
"System.lib"), which serves to recognize the last cycle before "STOP".

**Note**

**The mode selector switch position does not affect software start/stop!**
The position of the operating mode switch does not prevent the starting and
stopping of the PFC application from WAGO-I/O-*PRO*.

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

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

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

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

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

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

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

## 4.4.3    DIP Switch
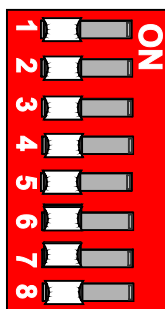


Figure 27: DIP Switch

The DIP switch is used to set the baud rate of the fieldbus controller and to set the DeviceNet<sup>TM</sup> station address (relating to DeviceNet<sup>TM</sup>, also called "MAC ID").

Settings are made by moving the slide switches to "ON" or "OFF".

The position of the individual slide switches only evaluated when turning on the fieldbus controller, i.e., changes are applied when the power supply for the fieldbus controller is turned OFF then ON again.

### 4.4.3.1    Baud Rate Setting

The baud rate is set using slide switches 7 and 8 of the DIP switch. 3 different baud rates are supported.

Table 20: Setting the Baud Rate via DIP Switch

| Baud rate | Slide switch 7 | Slide switch 8 |
|-----------|----------------|----------------|
| 125 kBaud (default) | OFF | OFF |
| 250 kBaud | ON | OFF |
| 500 kBaud | OFF | ON |
| not permitted | ON | ON |

**Example:**

Setting the baud rate to 250 kBaud
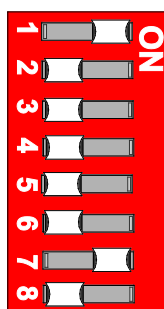


-   ON
-   OFF

Figure 28: Example with Baud rate Sent to 250 kBaud

### 4.4.3.2    Station address

The station address is set using slide switches 1 to 6.

The binary significance of the individual slide switches increases in the direction of the slide switch numbers. Slide switch 1 is used to set the lowest bit with a significance of $2^0$ and slide switch 6 to set the highest bit with a significance of $2^5$. If slide switch 1 is set to "ON", "MAC ID" 1 is set, if slide switch 1 and 4 are set to "ON", "MAC ID" 9 is set (($2^0 + 2^3$).

Station addresses in the range of 0 (all slide switches set to "OFF") to 63 (all slide switches set to "ON") can be set for the DeviceNet$^{TM}$ fieldbus nodes. The station address is set to 1 when delivered.

**Example:**

Setting the station address to 25.



- ON    $2^0 =$  1
- OFF
- OFF
- ON    $2^3 =$  8
- ON    $2^4 = 16$
- OFF
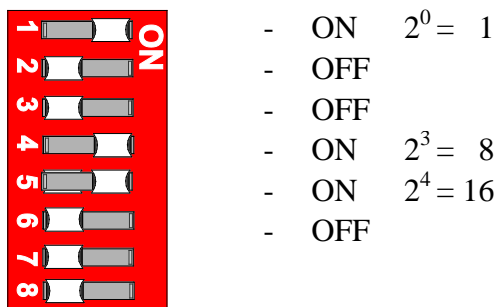
Figure 29: Example with Station Address Set to 25

## 4.5    Technical Data

### 4.5.1    Device Data

Table 21: Technical Data, Device Data

| Width | 51 mm |
|---|---|
| Height (from upper edge of DIN 35 rail) | 65 mm |
| Depth | 100 mm |
| Weight | approx. 195 g |
| Protection type | IP20 |

### 4.5.2    System Data

Table 22: Technical Data, System Data

| Number of I/O modules | 64 with scanner |
|---|---|
| Number of I/O points | approx. 6000 (master/slave dependent) |
| Transmission medium | Shielded Cu cable<br>Remote bus cable: 2 x 0.82 mm$^2$ + 2 x 1.7 mm$^2$<br>Drop cable: 2 x 0.2 mm$^2$ + 2 x 0.32 mm$^2$ |
| Bus segment length $_{max}$ | 100 m ... 500 m<br>(depending on baud rate/cable) |
| Network length $_{max}$ | acc. specification IEEE 802.3,<br>drop cables for all baud rates 6 m |
| Baud rate | 125 kBaud, 250 kBaud, 500 kBaud |
| Buscoupler connection | 5-pin male connector, Series 231 (*MCS*) |
| Protocols | DeviceNet™ |
| Programming | WAGO-I/O-*PRO* |
| IEC 61131-3 | IL, LD, FBD, ST, FC |
| Number of I/O modules | 64 |
| Input process image $_{max}$ | 1024 bytes |
| Output process image $_{max}$ | 1024 bytes |
| Input variables $_{max}$ | 512 bytes |
| Output variables $_{max}$ | 512 bytes |
| Configuration | via PC or PLC |
| Program memory | 128 Kbytes |
| Data memory | 64 Kbytes |
| Non-volatile memory (retain) | 8 Kbytes |
| Cycle time | < 3 ms for 1000 bit instructions/256 digital inputs/outputs |

*$^*)$ This value applies to brand new devices at an ambient temperature of 25 °C. The guaranteed buffer time for the real-time clock decreases with increasing temperature and operating time.*

## 4.5.3    DeviceNet<sup>TM</sup> Fieldbus

Table 23: Technical Data, DeviceNet<sup>TM</sup> Fieldbus

| DeviceNet<sup>TM</sup> characteristics | "Polled I/O Message Connection", "Strobed I/O Message Connection", "Change of State"/"Cyclic Message Connection", UCMM using function blocks, DeviceNet<sup>TM</sup> master can be programmed |
|---|---|

## 4.5.4    Power Supply

Table 24: Technical Data, Power Supply

| Voltage via power jumper contacts | 24 VDC (-15% ... +20%) |
|---|---|
| Current via power jumper contacts $_{max.}$ | 10A DC |
| Power supply efficiency$_{typ.}$ at nominal load | 87% |
| Internal current consumption | 350 mA at 5 V |
| Current consumption via - Power supply - CAN interface | < 500 mA at 24 V < 120 mA at 11 V |
| Total current for I/O modules | 1650 mA at 5 V |
| Electrical isolation | 500 V system/supply |

## 4.5.5    Accessories

Table 25: Technical Data, Accessories

| Pluggable connectors | Plug connectors 231-305/010-000/050-000 for *MCS* male connector (included) |
|---|---|
| Marking | Miniature WSB Quick marking system |
| PC software | WAGO-I/O-*PRO*, 759-333 |
| EDS Files | Download via http://www.wago.com |

## 4.5.6    Connection Type

Table 26: Technical Data – Field Wiring

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

Table 27: Technical Data – Power Jumper Contacts

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

Table 28: Technical Data – Data Contacts

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

## 4.5.7    Climatic Environmental Conditions

Table 29: Technical Data – Climatic Environmental Conditions

| | |
|---|---|
| Operating temperature range | 0 °C … 55 °C |
| Operating temperature range for components with extended temperature range (750-xxx/025-xxx) | −20 °C … +60 °C |
| Storage temperature range | −25 °C … +85 °C |
| Storage temperature range for components with extended temperature range (750-xxx/025-xxx) | −40 °C … +85 °C |
| Relative humidity | Max. 5 % … 95 % without condensation |
| Resistance to harmful substances | Acc. to IEC 60068-2-42 and IEC 60068-2-43 |
| Maximum pollutant concentration at relative humidity < 75 % | $SO_2 \leq 25$ ppm <br> $H_2S \leq 10$ ppm |
| Special conditions | Ensure that additional measures for components are taken, which are used in an environment involving: <br> – dust, caustic vapors or gases <br> – ionizing radiation |

## 4.5.8    Mechanical Strength acc. to IEC 61131-2

Table 30: Technical Data – Mechanical Strength acc. to IEC 61131-2

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

# 4.6    Approvals

---

**i**    *Information*

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

---

The following approvals have been granted to 750-806 fieldbus
coupler/controller:

CE    Conformity Marking

cUL<sub>US</sub>    cUL$_{US}$            UL508


ODVA    "Open DeviceNet Vendors Association" certified


The following Ex approvals have been granted to 750-806 fieldbus
coupler/controller:

TÜV 07 ATEX 554086 X

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

IECEx TUN 09.0001 X

Ex d I Mb
Ex nA IIC T4 Gc
Ex tc IIIC T135°C Dc

cUL$_{US}$            ANSI/ISA 12.12.01

Class I, Div2 ABCD T4


Korea Certification            MSIP-REM-W43-FBC750

Brasilian-    TUEV 12.1297 X
Ex        Ex nA IIC T4 Gc

WAGO®

The following ship approvals have been granted to 750-806 fieldbus coupler/controller:

ABS (American Bureau of Shipping)

Federal Maritime and Hydrographic Agency

BV (Bureau Veritas)

DNV (Det Norske Veritas)              Class B

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

KR (Korean Register of Shipping)

LR (Lloyd's Register)                 Env. 1, 2, 3, 4

NKK (Nippon Kaiji Kyokai)

PRS (Polski Rejestr Statków)

RINA (Registro Italiano Navale)

# 4.7    Standards and Guidelines

750-806 meets the following requirements on emission and immunity of interference:

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

EMC CE-Emission of interference       acc. to EN 61000-6-3

EMC marine applications-Immunity
to interference                                       acc. to Germanischer Lloyd

EMC marine applications-Emission
of interference                                       acc. to Germanischer Lloyd

# 5    Mounting

## 5.1    Installation Position

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

> **Note**
>
> → **Use an end stop in the case of vertical mounting!**
> In the case of vertical assembly, an end stop has to be mounted as an additional safeguard against slipping.
> WAGO order no. 249-116    End stop for DIN 35 rail, 6 mm wide
> WAGO order no. 249-117    End stop for DIN 35 rail, 10 mm wide

## 5.2    Overall Configuration

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

**Examples:**

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

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

**Exception:**

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

> **NOTICE**
>
> **Observe maximum total length of a fieldbus node!**
> The maximum total length of a fieldbus node without fieldbus coupler/controller and without using a 750-628 I/O Module (coupler module for internal data bus extension) may not exceed 780 mm.
> Also note the limitations of individual fieldbus couplers/controllers.

> ## Note
>
> **Increase the total length using a coupler module for internal data bus extension!**
> You can increase the total length of a fieldbus node by using a 750-628 I/O Module (coupler module for internal data bus extension). For such a configuration, attach a 750-627 I/O Module (end module for internal data bus extension) after the last I/O module of a module assembly. Use an RJ-45 patch cable to connect the I/O module to the coupler module for internal data bus extension of another module block.
> This allows you to segment a fieldbus node into a maximum of 11 blocks with maximum of 10 I/O modules for internal data bus extension.
> The maximum cable length between two blocks is five meters.
> More information is available in the manuals for the 750-627 and 750-628 I/O Modules.

# 5.3    Mounting onto Carrier Rail

## 5.3.1    Carrier Rail Properties

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

---

### NOTICE

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

---

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

- The material must be non-corrosive.

- Most components have a contact to the carrier rail to ground electro-magnetic disturbances. In order to avoid corrosion, this tin-plated carrier rail contact must not form a galvanic cell with the material of the carrier rail which generates a differential voltage above 0.5 V (saline solution of 0.3 % at 20°C).

- The carrier rail must optimally support the EMC measures integrated into the system and the shielding of the I/O module connections.

- A sufficiently stable carrier rail should be selected and, if necessary, several mounting points (every 20 cm) should be used in order to prevent bending and twisting (torsion).

- The geometry of the carrier rail must not be altered in order to secure the safe hold of the components. In particular, when shortening or mounting the carrier rail, it must not be crushed or bent.

- The base of the I/O components extends into the profile of the carrier rail. For carrier rails with a height of 7.5 mm, mounting points are to be riveted under the node in the carrier rail (slotted head captive screws or blind rivets).

- The 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 31: WAGO DIN Rail

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

# 5.4    Spacing

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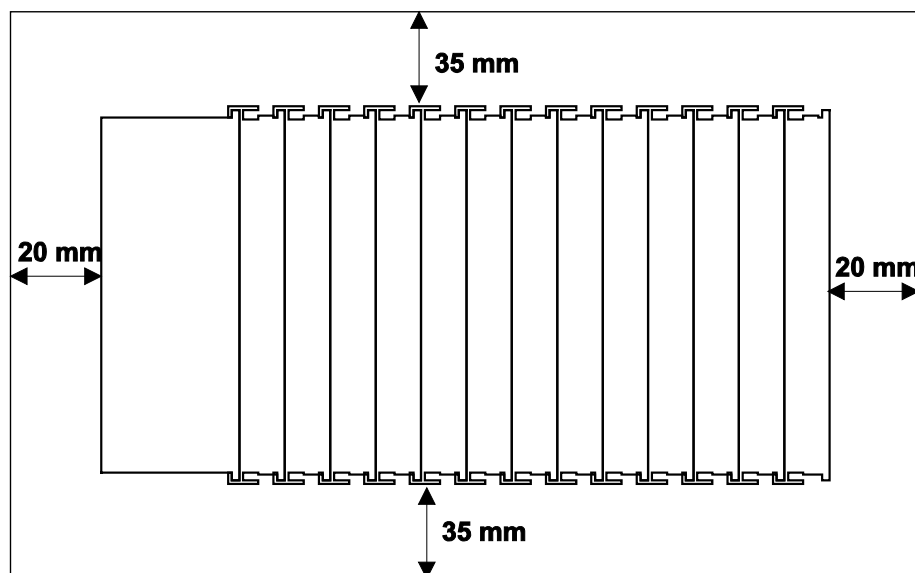


Figure 30: Spacing

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

## 5.5      Mounting Sequence

Fieldbus couplers/controllers and I/O modules of the WAGO-I/O-SYSTEM
750/753 are snapped directly on a carrier rail in accordance with the European
standard EN 50022 (DIN 35).

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

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

### ⚠ CAUTION

**Risk of injury due to sharp-edged blade contacts!**
The blade contacts are sharp-edged. Handle the I/O module carefully to prevent
injury.

### NOTICE

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

### Note

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

## 5.6    Inserting and Removing Devices

> **NOTICE**
>
> **Perform work on devices only if they are de-energized!**
> Working on energized devices can damage them. Therefore, turn off the power
> supply before working on the devices.

## 5.6.1    Inserting the Fieldbus Coupler/Controller

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

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

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

With the fieldbus coupler/controller snapped in place, the electrical connections for the data contacts and power contacts (if any) to the possible subsequent I/O module are established.
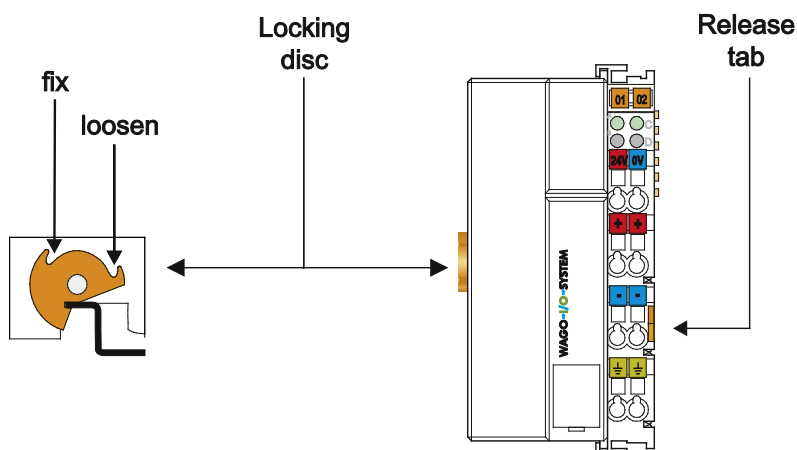
Figure 31: Release Tab Standard Fieldbus Coupler/Controller (Example)

## 5.6.2    Removing the Fieldbus Coupler/Controller

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

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

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

### 5.6.3    Inserting the I/O Module

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



Figure 32: Insert I/O Module (Example)

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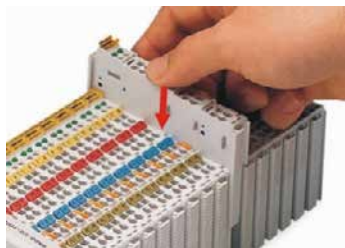


Figure 33: Snap the I/O Module into Place (Example)

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

## 5.6.4    Removing the I/O Module

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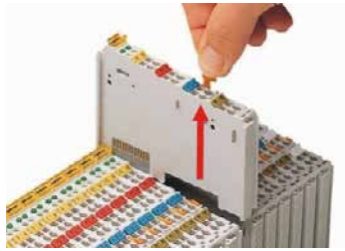


Figure 34: Removing the I/O Module (Example)

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

# 6        Connect Devices

## 6.1      Data Contacts/Internal Bus

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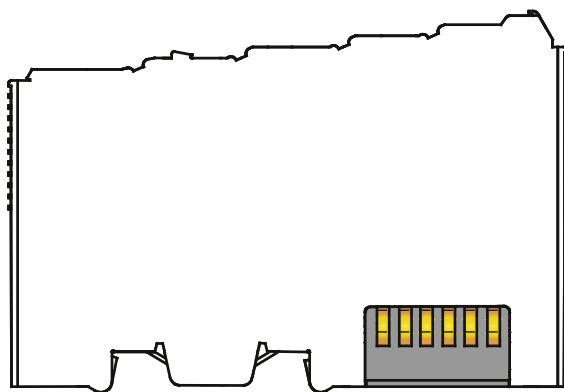


Figure 35: Data Contacts

---

**NOTICE**

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

---

**NOTICE**

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

## 6.2     Power Contacts/Field Supply

> ⚠ **CAUTION**
>
> **Risk of injury due to sharp-edged blade contacts!**
> The blade contacts are sharp-edged. Handle the I/O module carefully to prevent
> injury.

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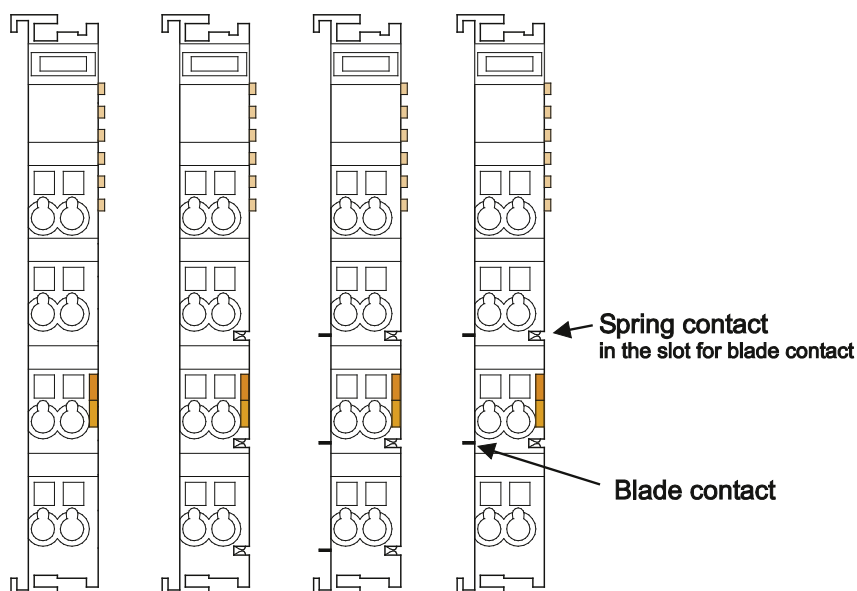


Figure 36: Example for the Arrangement of Power Contacts

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

## 6.3       Connecting a Conductor to the CAGE CLAMP®

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

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

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

**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.     For opening the CAGE CLAMP® insert the actuating tool into the opening above the connection.

2.     Insert the conductor into the corresponding connection opening.

3.     For closing the CAGE CLAMP® simply remove the tool. The conductor is now clamped firmly in place.
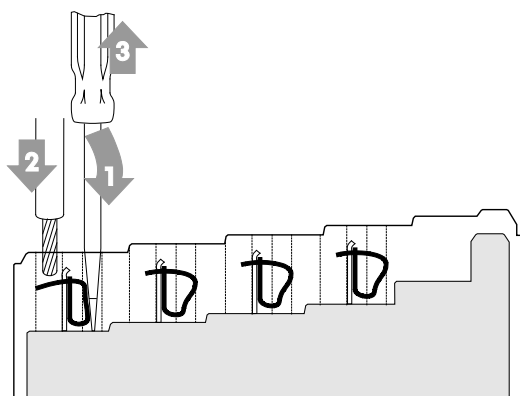


Figure 37: Connecting a Conductor to a CAGE CLAMP®

# 7        Function Description

## 7.1     Operating System

### 7.1.1    Run-up

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

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

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

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

> **Information**
>
> **More information about the LED Signaling**
> Read the detailed description for the evaluation of the displayed LED state in the section "Diagnostics" > … > "LED Signaling".

### 7.1.2    PFC Cycle

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

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

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

The inputs, outputs and timer values are not updated while the PFC program is being processed. Updating is performed only as defined at the end of the PFC program. As a result, it is not possible to wait on an event from the process or a set period to expire while a loop is in progress.
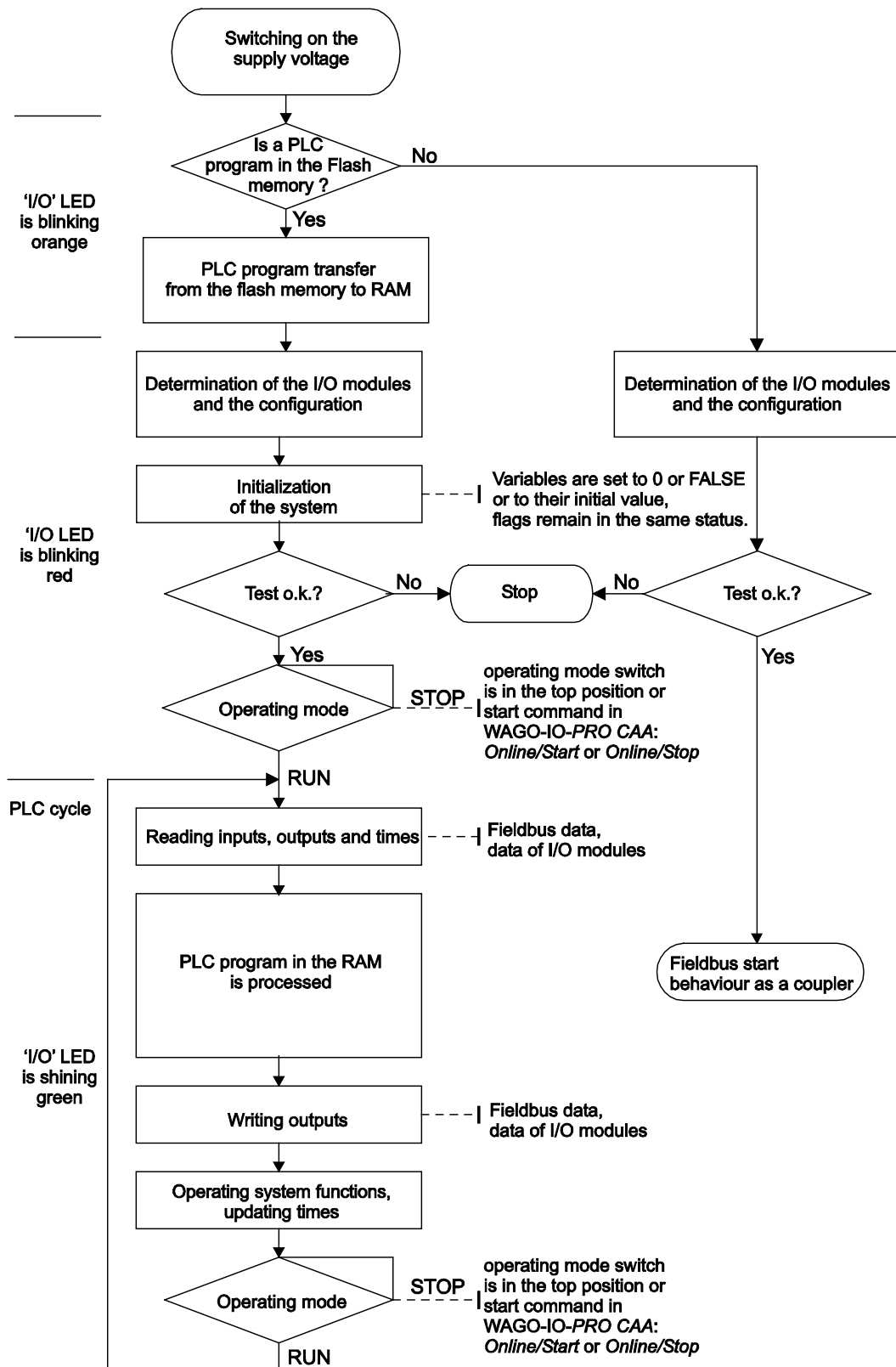
Figure 38: Run-up of the Fieldbus Controller

## 7.2        Process Data Architecture

### 7.2.1     Basic Structure

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

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

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

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

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

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

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

First, all the byte-oriented (analog) IO modules are filed in the process image, then the bit-oriented (digital) IO modules. The bits of the digital modules are grouped into bytes. If the amount of digital information exceeds 8 bits, the controller automatically starts with a new byte.

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

# 7.3    Data Exchange

Objects are used for exchange of process data for the DeviceNet™- fieldbus controller.

For access from the network to the individual objects, connections between the required subscribes must first be established and connection objects set up or activated.

The fieldbus controller can be addressed from the "Unconnected Message Manager Port" (UCMM port).

The UCMM port allows dynamic connections via one or more connections from one or more clients.

The "Assembly Object" specifies the structure of the objects for data transmission. With the "Assembly Object", I/O data can be combined into blocks, for example, and sent via a single message connection. Creating blocks requires less traffic on the network.

There is a distinction between input and output assemblies. An input assembly reads in data from the application via network or produces data on the network.

An output assembly writes data to the application or pulls data from the network. In the fieldbus controller, various assembly instances are pre-programmed (static assembly).



**Information**

**Additional Information**
The assembly instances for static assembly are described in the section "Fieldbus Communication" > … > "Process Data and Diagnostic Status" > "Assembly Instances."

The dynamic assembly can also be used from the static assembly at the fieldbus controller. In the dynamic assembly, assembly instances can be created in which process data can be compiled from various application objects as required.



**Information**

**Additional Information**
Information on the dynamic assembly is available in section "Programming PFC with WAGO-I/O-*PRO"* > "Special DeviceNet™ Characteristics of the Fieldbus Controller" > "Dynamic Assembly".

### 7.3.1    Communication Interfaces

For the data exchange, the fieldbus controller has essentially three interfaces:

- interface to the fieldbus (fieldbus master)
- PLC function of the fieldbus controller (CPU)
- interface to the I/O modules.

There is a data exchange between: fieldbus master and the I/O modules, PLC function of the fieldbus controller (CPU) and the I/O modules and between the fieldbus master and PLC function of the PFC (CPU).

The PLC functionality of the PFC (CPU) accesses the data via IEC 61131-3 variables.

Access from the fieldbus side is fieldbus specific.

### 7.3.2    Memory Space

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

The fieldbus controller has an additional memory area for the figure of PFC variables defined according to IEC 61131-3. In this extended memory area (each word 256 ... 511), the PFC variables are mapped behind the physical process data.

Access by the PLC to process data is made independently from the fieldbus system in all WAGO fieldbus controllers; access is always conducted through an application-related IEC-61131-3 program.
How the data is accessed from the fieldbus side depends on the fieldbus however.
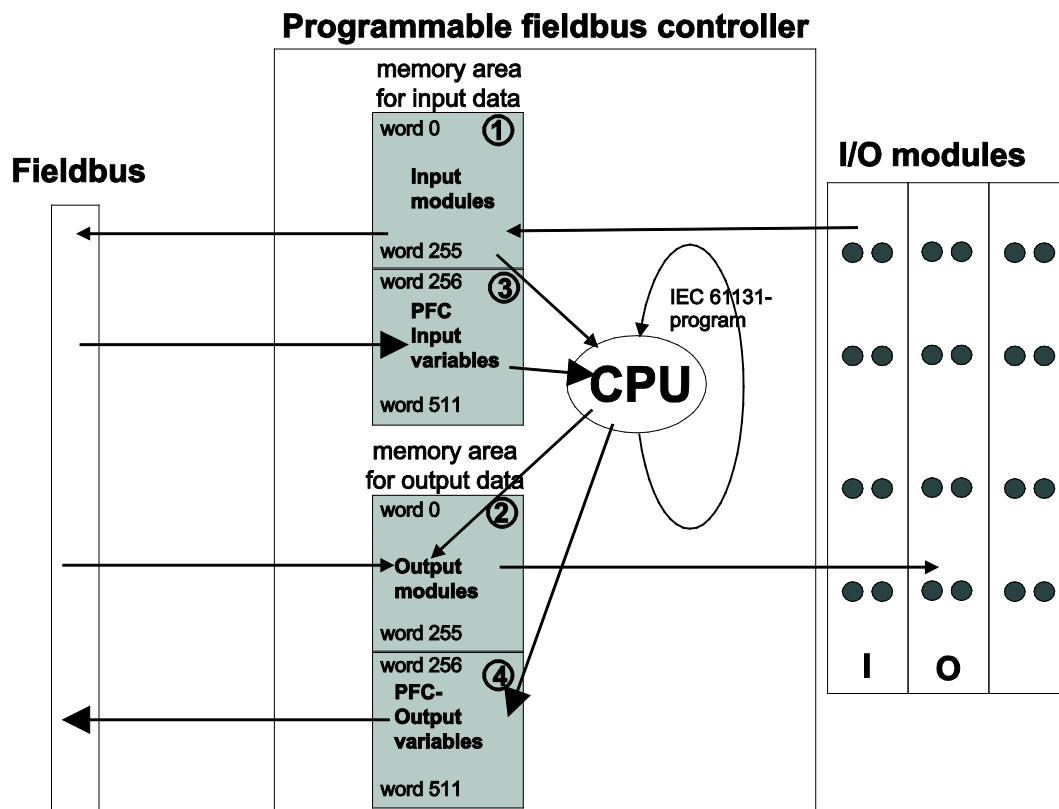
Figure 39: Memory Areas

In the memory space word 0 … 255, the fieldbus controller process image contains the physical data of the bus modules.

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

2       In the same manner, writing on the output modules is possible from the CPU and from the fieldbus side. The value of the fieldbus master is put out on the output while writing on an output.

The PFC variables (fieldbus variables) are filed in the memory space Word 256 … 511 of the process image.

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

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

## Note

**Avoid writing to one output at the same time!**
Avoid simultaneous writing to an output either by using instance 11 of the static assemblies (see section "Fieldbus Communication" > … > "Additional Assembly Instances 10 and 11") or by using the dynamic assembly (see section "Programming PFC with WAGO-I/O-PRO" > "Special DeviceNet™ Characteristics of the Fieldbus Controller" > … > "Dynamic Assembly").

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

- **Data memory (64 kByte)**
  The data memory is a volatile RAM memory for creating variables that are not required for communication with the interfaces, but rather for internal processing procedures, such as calculation of results.

- **Program memory (128 kByte)**
  The IEC-61131-3 program is stored in the program memory. The code memory is a Flash ROM. When power is switched on, the program is transferred from the flash to the RAM memory. After error-free run-up, the PFC cycle starts with the mode selector switch at the top position, or on the Start command from the WAGO-I/O-*PRO*.

- **NOVRAM Remanent memory (8 kByte)**
  The remanent memory is a non-volatile memory; i.e., all values of flags and variables, that are explicitly defined by "var retain", are retained even after a loss of power. Memory management is performed automatically. The 8 KB memory area is used jointly for flags and retain variables.

## Note

**Markers are only remanent under "var retain"!**
Please note that the bit memory is only retentive if you have declared it as such under "var retain".
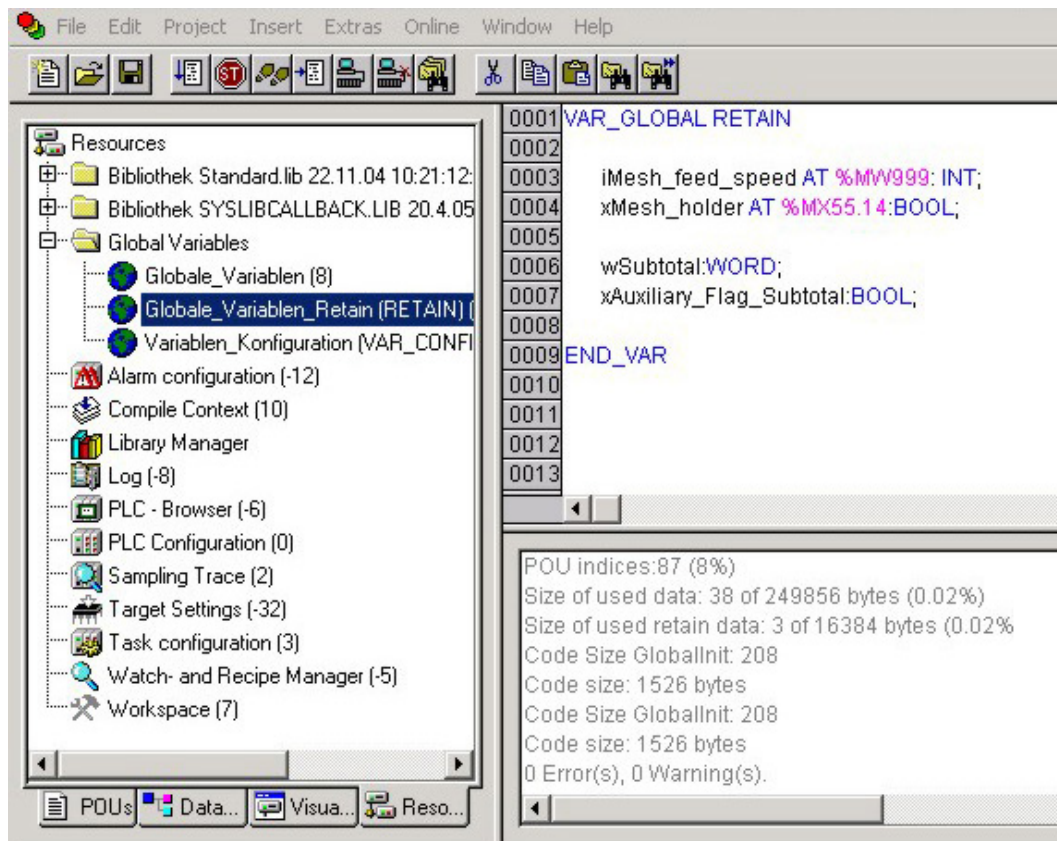
Figure 40: Example Declaration of Remanent Flags by "var retain"

This breakdown can be varied (see following explanation).

> **Note**
>
> → **NOVRAM memory allocation can be changed in WAGO-I/O-*PRO*!**
> The breakdown of the NOVRAM can be modified when required in the programming software WAGO-I/O-*PRO*/Register "Resources"/Dialog window "Target system settings".
> The start address for the flag and retain range is specified by default as 16#100000 for the fieldbus controller. The range sizes and the start address can be varied, however.
> If you use the default values, the flag and retain range will overlap. Depending on the declaration in the program, the range is then used for flag or retain variables.
> In order to prevent an overlapping of the ranges, you can specify 16#102000 as the retain start address, for example. In this case, the flags will be stored first (starting with 16#100000) and then the retain variables (starting with 16#102000).

# 7.3.3    Addressing

## 7.3.3.1    Fieldbus-Specific Addressing

After turning on the power supply, the assembly object combines data from the process image. As soon as a connection is established, a DeviceNet<sup>TM</sup> master (scanner) can address the data with "class", "instance" and "attribute" and then access it or read and/or write via I/O connections.

Data mapping depends on the assembly instance selected or on application-specific determination with the dynamic assembly.

---

**Information**

**Additional Information**
The assembly instances for static assembly are described in the section "Fieldbus Communication" > … > "Assembly Instances".

---

**Information**

**Additional Information**
Information on the dynamic assembly is available in section "Programming PFC with WAGO-I/O-*PRO"* > "Dynamic Assembly".

---

**Information**

**Additional Information**
For the number of input and output bits or bytes of the individual I/O modules, please refer to their corresponding descriptions.

---
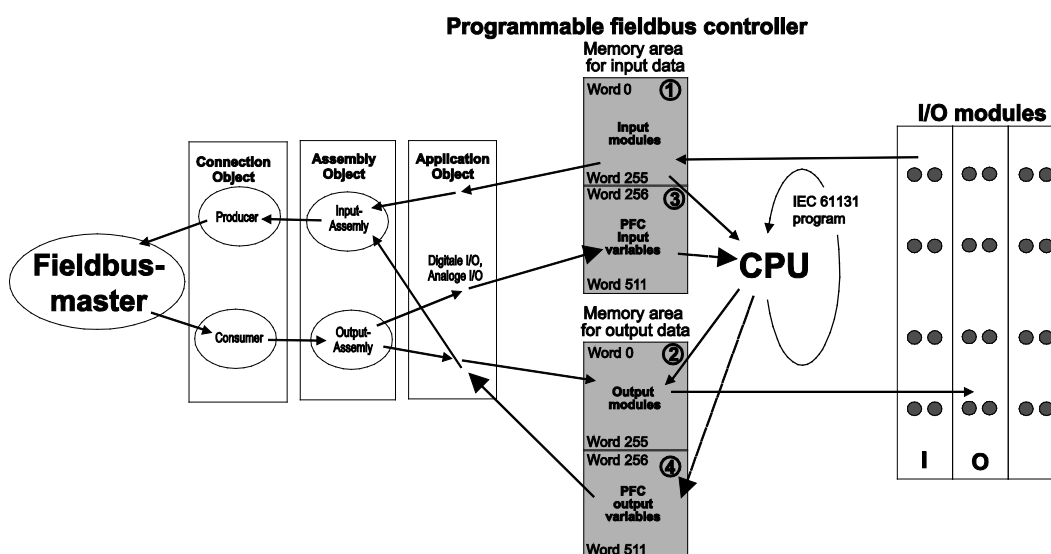


Figure 41: Memory Areas and Addressing

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

### Example of a static assembly:

On delivery of the fieldbus controller, the default setting for the static assembly is:

- Output1 (I/O assembly instance 1)
- Input1    (I/O assembly instance 4)

### Example setup of the fieldbus node:

1. DeviceNet™ fieldbus controller (750-806)
2. 4-channel digital input module (e.g., 750-402)
3. 4-channel digital output module (e.g., 750-504)
4. 2-channel analog output module with 2 bytes per channel (e.g., 750-552)
5. 2-channel analog input module with 2 bytes per channel (e.g., 750-456)
6. End module (750-600)

### Input process image:

Standard process data, input image (Assembly Class, Instance 4)

Table 32: Input Process Image

|        | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| **Byte 0** | Low byte channel 1 ||||||||
| **Byte 1** | High byte channel 1 ||||||||
| **Byte 2** | Low byte channel 2 ||||||||
| **Byte 3** | High byte channel 2 ||||||||
| **Byte 4** | unused ||||  DI04[1] | DI03[1] | DI02[1] | DI01[1] |
| **Byte 5** | DS08[2] | DS07[2] | DS06[2] | DS05[2] | DS04[2] | DS03[2] | DS02[2] | DS01[2] |

[1] DI = Digital input

[2] DS = Diagnostic status (The last byte in the input process image is the diagnostic status byte, DS01 … DS08, see also object 0x64/Instance 1/Attr. 5)

   DS01 =1: Internal bus error (0x01)
   DS02 =1: Coupler configuration error (0x02)
   DS04 =1: Module diagnostics (0x08)
   DS08 =1: Fieldbus error (0x80)

**Output process image:**

Standard process data, output image (Assembly Class, Instance 1)

Table 33: Output Process Image

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Byte 0** | Low byte channel 1 | | | | | | | |
| **Byte 1** | High byte channel 1 | | | | | | | |
| **Byte 2** | Low byte channel 2 | | | | | | | |
| **Byte 3** | High byte channel 2 | | | | | | | |
| **Byte 4** | unused | | | | DO04[1] | DO03[1] | DO02[1] | DO01[1] |

[1] DO = Digital output

## 7.3.3.2 Absolute Addressing

The internal CPU of the fieldbus controller has direct access to the data via absolute addresses. The addressing starts with address 0 for both inputs and outputs. The respective addresses for bits, bytes and double words (DWord) are derived from the word addresses.

The structure of the process image is described in section "Functional Description" > … > "Standard Design".

Addressing occurs according to the following structure:

| **Input data** | $\%IW0$ | |
| | … | word-oriented data |
| | $\%IW_n$ | |
| | | |
| | $\%I_{n+1}$ | |
| | … | bit-oriented data |
| | $\%I_{n+m}$ | |

| **Output data** | $\%QW0$ | |
| | … | word-oriented data |
| | $\%QW_n$ | |
| | | |
| | $\%Q_{n+1}$ | |
| | … | bit-oriented data |
| | $\%Q_{n+m}$ | |

### 7.3.3.3    Calculate Addresses

The word address is the basses for calculating addresses.

| | |
|---|---|
| **Bit address** | Word address .0 to .15 |
| **Byte address** | 1st byte: 2 x word address<br>2nd byte: 2 x word address + 1 |
| **DWord address** | lower section: Word address (even numbers) / 2<br>upper section: Word address (odd numbers) / 2,<br>rounded off |

### 7.3.3.4    Address Range for I/O Module Data

Table 34: Address Range for I/O Module Data

| Data width | Address range for I/O module data | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Bit** | 0.0<br>…<br>0.7 | 0.8<br>…<br>0.15 | 1.0<br>…<br>1.7 | 1.8<br>…<br>1.15 | … | 254.0<br>...<br>254.7 | 254.8<br>...<br>254.15 | 255.0<br>...<br>255.7 | 255.8<br>...<br>255.15 |
| **Byte** | 0 | 1 | 2 | 3 | … | 508 | 509 | 510 | 511 |
| **Word** | **0** | | **1** | | **…** | **254** | | **255** | |
| **DWord** | 0 | | | | … | 127 | | | |

### 7.3.3.5    Address Range for Fieldbus Variables

Table 35: Address Range for Fieldbus Variables

| Data width | Address range for fieldbus variables | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Bit** | 256.0<br>…<br>256.7 | 256.8<br>…<br>256.15 | 257.0<br>…<br>257.7 | 257.8<br>…<br>257.15 | … | 510.0<br>...<br>510.7 | 510.8<br>...<br>510.15 | 511.0<br>...<br>511.7 | 511.8<br>...<br>511.15 |
| **Byte** | 512 | 513 | 514 | 515 | … | 1020 | 1021 | 1022 | 1023 |
| **Word** | **256** | | **257** | | **…** | **510** | | **511** | |
| **DWord** | 128 | | | | … | 255 | | | |

### 7.3.3.6    Address Range for Flags

Table 36: Address Flags

| Data width | Address range for flags | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Bit** | 0.0<br>…<br>0.7 | 0.8<br>…<br>0.15 | 1.0<br>…<br>1.7 | 1.8<br>…<br>1.15 | … | 4094.0<br>…<br>4094.7 | 4094.8<br>…<br>4094.15 | 4095.0<br>…<br>4095.7 | 4095.8<br>…<br>4095.15 |
| **Byte** | 0 | 1 | 2 | 3 | … | 8188 | 8189 | 8190 | 8191 |
| **Word** | **0** | | **1** | | **…** | **4094** | | **4095** | |
| **DWord** | 0 | | | | … | 2047 | | | |

All flags are non-volatile (retain).

### 7.3.3.7    Example for Absolute Addresses

| Data width | Inputs | | | |
|---|---|---|---|---|
| Bit | %IX14.0 ... 15 | | %IX15.0 ... 15 | |
| Byte | %IB28 | %IB29 | %IB30 | %IB31 |
| **Word** | **%IW14** | | **%IW15** | |
| DWord | %ID7 | | | |

| Data width | Outputs | | | |
|---|---|---|---|---|
| Bit | %QX5.0 ... 15 | | %QX6.0 ... 15 | |
| Byte | %QB10 | %QB11 | %QB12 | %QB13 |
| **Word** | **%QW5** | | **%QW6** | |
| DWord | %QD2 (top section) | | %QD3 (bottom section) | |

| Data width | Flag | | | |
|---|---|---|---|---|
| Bit | %MX11.0 ... 15 | | %MX12.0 ... 15 | |
| Byte | %MB22 | %MB23 | %MB24 | %MB25 |
| **Word** | **%MW11** | | **%MW12** | |
| DWord | %MD5 (top section) | | %MD6 (bottom section) | |

The character "X" for single bits can be omitted, e.g., %I14.0, %Q6.10, %M11.7.

# 8     Commissioning

This section shows a step-by-step procedure for starting up exemplarily 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 DeviceNet^TM fieldbus node with a non-networked computer under Windows.

Three work steps to start up are described. Theses work steps are described in the corresponding following sections.

- **Connecting PC and fieldbus nodes**

- **Setting the "MAC ID" and baud rate**

- **Configuring static and dynamic assemblies**

After the topics specified above, you can find instructions for programming the fieldbus controller with WAGO-I/O-*PRO*.

## 8.1      Connecting Client PC and Fieldbus Nodes

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

2.  Use a fieldbus cable to connect the DeviceNet™ fieldbus node to the
    DeviceNet™ fieldbus card in your PC.

    24V power is supplied to the fieldbus via connections V+, V- of the 5-pin
    fieldbus connector (Series 231 *(MCS)*) from an external fieldbus power supply
    unit.

3.  Turn the operating voltage on.

4.  Turn on your PC.

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

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

---

### *Information*

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

---

## 8.2 Setting the DeviceNet™ Station Address and Baud Rate

1. Set the required DeviceNet™ station address ("MAC ID") using slide switches 1 … 6 of the DIP switch.

2. Set the required baud rate using slide switches 7 and 8.

**Example:**

Setting the station address "MAC ID" 4 (DIP 3 = ON) and baud rate 500 kBaud (DIP 7 = OFF, DIP 8 = ON).

| | |
|---|---|
| OFF | |
| OFF | |
| **ON** $(2^2 = 4)$ | Station address |
| OFF | |
| OFF | |
| OFF | |
| **OFF** | Baud rate |
| **ON** | |

Figure 42: Example of DIP Switch Settings

3. Then switch the fieldbus controller power supply on.

## 8.3      Configuring Static and Dynamic Assemblies

In this example, RSNetWorx software version 21 from Allen-Bradley and
RSLogix 5000 is used with a "1769 SDN scanner module".

The inputs are mapped via the static assembly and the outputs mapped via the
dynamic assembly.

The sample node is configured with the following I/O modules:

| Fieldbus controller 750-806 | Digital input module 750-402 | Digital output module 750-516 | Digital output module 750-516 | Digital output module 750-516 | Analog input module 750-467 | Analog output module 750-550 | End module 750-600 |
|---|---|---|---|---|---|---|---|

1.      Launch the "RSNetWorx" configuration software.

2.      To load the EDS file "WAGO 750-806_2_8.eds", click the "EDS Wizard"
        menu item in the "Tools" menu and follow the instructions.

**Information**

**Additional Information**
The EDS file "WAGO 750-806_2_8.eds" can be downloaded from the WAGO
website http://www.wago.com in the Download area.

After loading the EDS file in "RSNetWorx", you can start configuring your
network:

3.      Click the "Communication Adapter" folder in the tree structure on the left
        side of the screen.

        Various subfolders are listed.

4.      In the subfolders listed, select the corresponding scanner (DeviceNet™
        Master) that is in your network (for this example, select "1769 SDN Scanner
        Module").

5.      Double-click to apply the scanner selected or drag & drop the selection to
        the right side of the screen.

6.      The scanner selected appears as an icon on the right side of the screen.

7.    Now select the DeviceNet™ Fieldbus Controller (750-806) in the tree
      structure under the "Communication Adapter" folder.

8.    Double-click to apply or drag & drop the selection to the right side of the
      screen.

      The fieldbus controller is added as a second icon on the right side of the
      screen.

The correct configuration settings for incoming and outgoing data (TX/RX) are
required for proper function with the DeviceNet™ network. The TX/RX
configuration must match the node configuration.

9.    For the entries in the "RX" and "TX" fields, add up all input bits/bytes and
      all output bits/bytes. Individual bits are always amended to full bytes.

From the fieldbus master view, the sample node has the following data
configuration:

Table 37: Data Configuration for Sample Nodes

| I/O module | RX | TX |
|---|---|---|
| 750-806   Fieldbus Controller DeviceNet<sup>TM</sup> | 1 byte input status | |
| 750-402   4-channel digital input module | 4 bits input | |
| 750-402   4-channel digital input module | 4 bits input | |
| 750-516   4-channel digital output module | | 4 bits output |
| 750-516   4-channel digital output module | | 4 bits output |
| 750-516   4-channel digital output module | | 4 bits output |
| 750-467   2-channel analog input module | 4 bytes input | |
| 750-550   2 channel analog output module | | 4 bytes output |
| 750-600   End module | - | - |
| PFC fieldbus input variables | | 0 bytes input |
| PFC fieldbus output variables | 4 bytes output | |
| **Total** | **10 bytes** | **6 bytes** |

→ **Note**

**Make sure that TX/RX are configured correctly!**
The PFC output variables are defined from the perspective of the programmable
fieldbus controller. This involves input variables from the perspective of the
DeviceNet™ fieldbus. These are added to the RX settings. Accordingly, the PFC
input variables for access from a PLC are output variables for the fieldbus.
Therefore, these are added to the TX settings:
PLC input variable = PFC output variable
PFC input variable = PLC output variable

**Fieldbus**                    **Programmable**
                                **fieldbus controller**

PLC                             PFC
input                           input
variables                       variables

PLC                             PFC
output                          output
variables                       variables
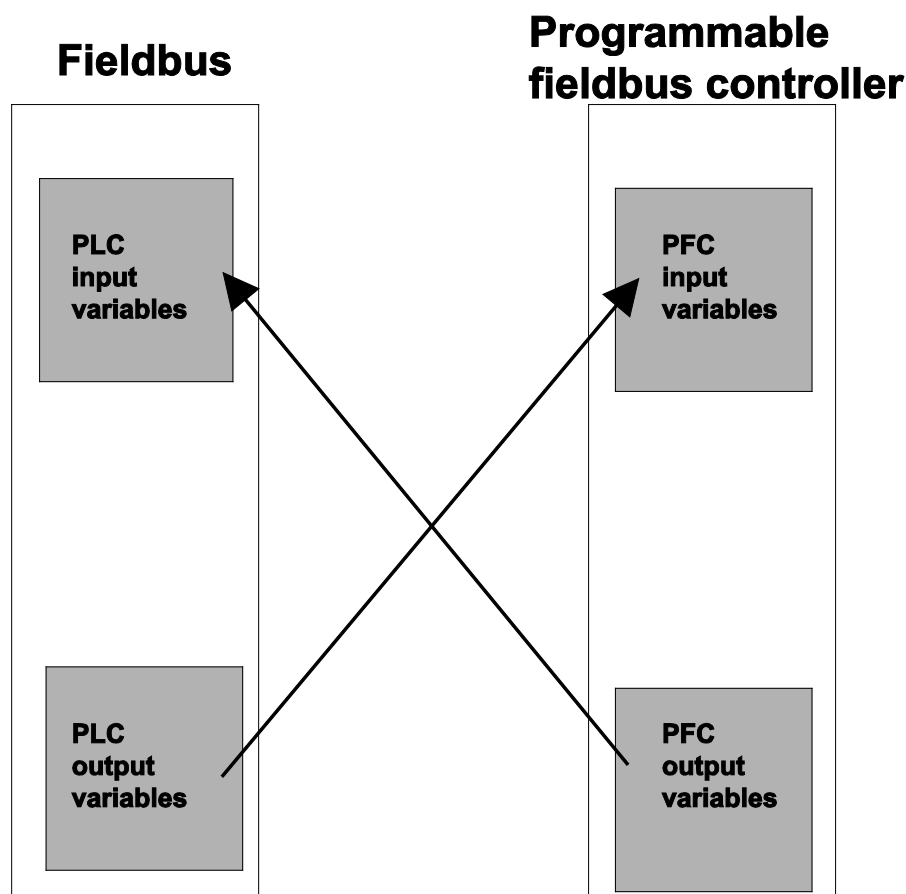
Figure 43: Correlation Between PLC Variables and PFC Variables

The DeviceNet™ Master (scanner) should have access to the physical inputs and to 4 bytes of PFC output variables in this example.

For the static assembly for the TX configuration of the scanner, 4 bytes of PFC output variables are added to the quantity of input data.

10. To parameterize the fieldbus controller, double-click the fieldbus node 750-806 icon.

    On the "General" table, you can assign the required address to the fieldbus controller.

11. Click in the input window for the address and enter the address according to the address set on the DIP switch of the fieldbus controller.

    On the "Configuration" tab, you can enter the RX/TX configuration.

12. Leave the value for "ID#37 PLC Fieldbus Input Variables" at "0". For "ID#38 Number of PLC Fieldbus Input Variables", enter "4".

13. Click the **[OK]** button to confirm the setting.

14. To start the configuration, double-click the scanner icon.

    The "1769-SDN Scanner Module" dialog window appears.

15.    Select the "Scan List" tab.

16.    Click the [→] button to move the DeviceNet<sup>TM</sup> Fieldbus Controller (750-806) from the "Available Devices" window on the left side to the "Scan List" window.

17.    Click the "Edit I/O Parameters..." button.

18.    Activate the Poll function by selecting the "Polled" check box.

     You can now enter the "Polled Produced Data length" and "Polled Consumed Data length".

19.    Enter 6 bytes in the "Input Size" dialog box.

     These are receiving bytes for the inputs.

20.    Enter 4 bytes for the PFC input variables in the "Output Size" dialog box. The number of bytes results from the following definitions in the dynamic assembly for the outputs. It is defined that only the PFC input variables and no physical outputs should be described by the master.

21.    Then click the **[OK]** button to apply the parameters.

     A window appears noting that some input/output data are not mapped.

22.    Click **[Yes]** to confirm the prompt that you want to continue.

23.    Select the "Input" tab in the "1769-SDN Scanner Module" dialog window.

24.    Now click the "Automap" button to display the mapping.

     All bits are mapped as digital inputs.
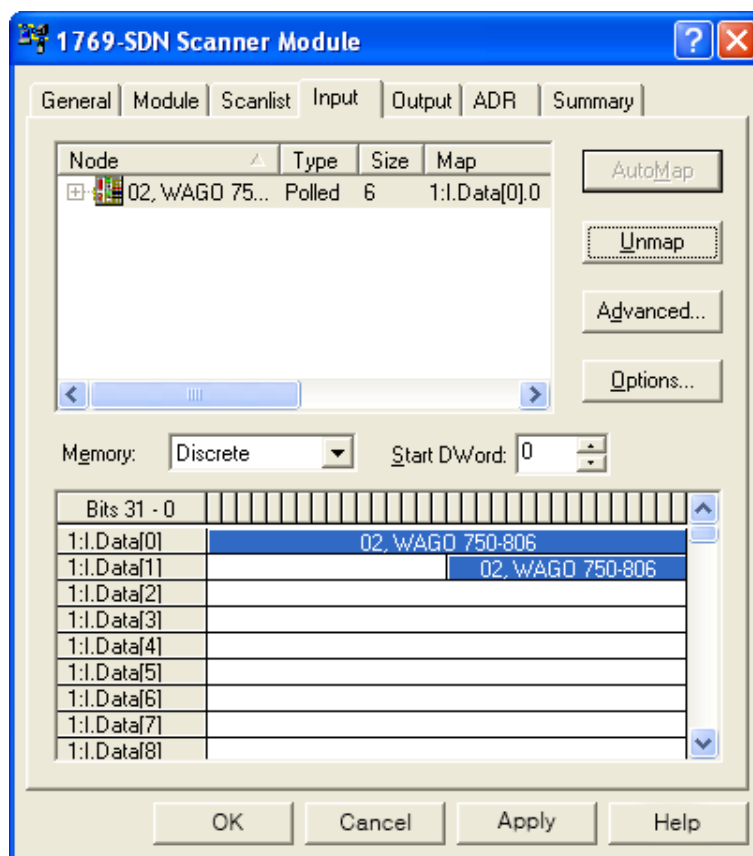
Figure 44: Mapping of Inputs as Digital Inputs

Table 38: Mapping of Inputs

| Bits | Figure |
|------|--------|
| I:1.0 | 1 word reserved for scanner module |
| I:1.1 | 1 word analog input channel 1 |
| I:1.2 | 1 word analog input channel 2 |
| I:1.3 | 1 byte status | 1 byte digital input |
| I:1.4 | 1 word PLC input variable 1 (or PFC output variable 1) |
| I:1.5 | 1 word PLC input variable 2 (or PFC output variable 2) |

25.    Select the "Output" tab in the "1769-SDN Scanner Module" dialog window.

26.    Now click the "Automap" button to display the mapping.

        All bits are mapped as digital outputs.
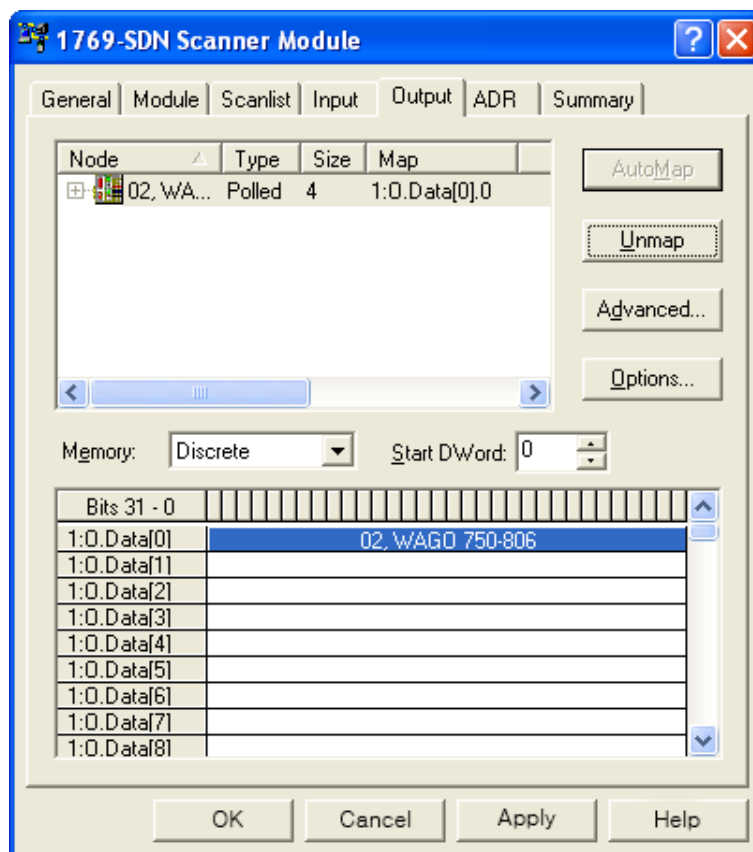
Figure 45: Mapping Outputs as Digital Outputs

Table 39: Mapping of Inputs

| Bits  | Figure |
|-------|--------|
| O:1.0 | 1 word reserved for scanner module |
| O:1.1 | 1 word PLC output variable 1 (or PFC input variable 1) |
| O:1.2 | 1 word PLC output variable 2 (or PFC input variable 2) |

With the dynamic assembly, data are mapped that should be transmitted via the fieldbus. The data are saved as classes, instances and attributes.

27. Click the icon for the fieldbus controller 750-806 in the graphical representation so that the icon is highlighted.
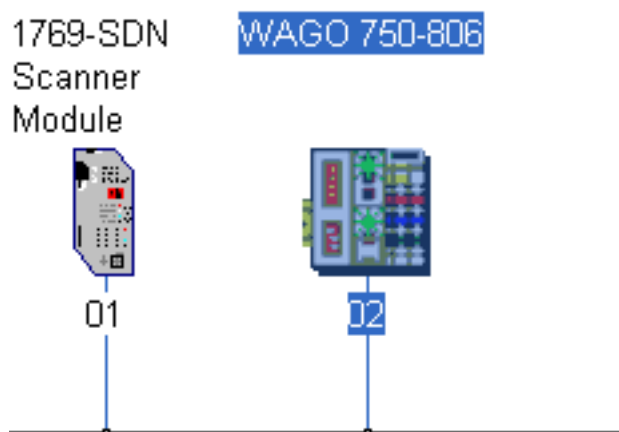


Figure 46: Selecting the fieldbus controller 750-806

26.    Then click the **Class Instance Editor...** menu item in the "Device" menu.
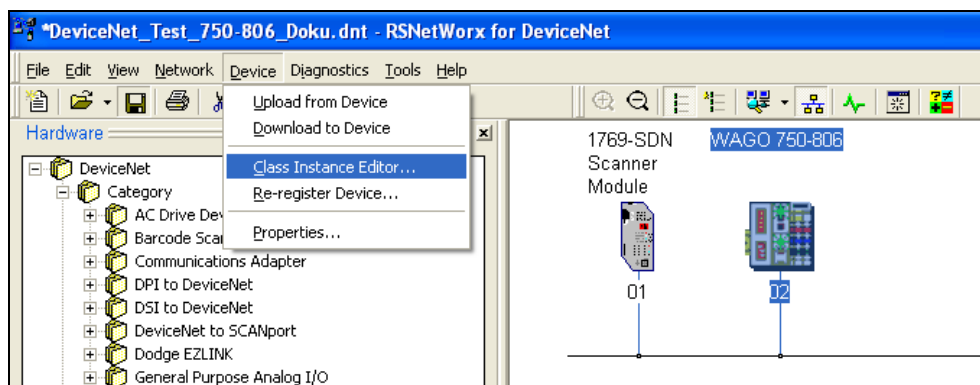


Figure 47: Opening the "Class Instance Editor"

A window opens with the following warning:
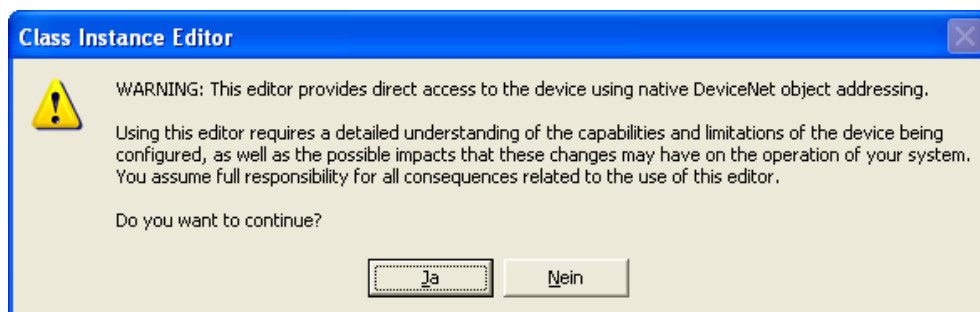


Figure 48: Warning



**Select a uniform number format!**
The editor changes parameters in the fieldbus controller. Therefore, consistently enter all data as hexadecimal or decimal.
If the number format of the data is not uniform, it can lead to data loss or cause the fieldbus controller to fail completely.

28.    Click **[Yes]** to confirm the warning.

The "Class Instance Editor" dialog window appears.

29.    Select the "Create" service in the "Description" dialog box and enter the following values in the dialog boxes for the "Object Address":

- "Class"        → "4"
- "Instance"     → "0"
- "Attribute"    → "1"

→ **Note**

**The [ENTER] closes the dialog window!**
Do not click **[ENTER]** because the dialog window closes and has to be opened
again.

30. Click the **[Execute]** button to create the instance for the dynamic assembly.
    If the instance is created successfully, the fieldbus node sends instance
    number "64 00" (hexadecimal notation).
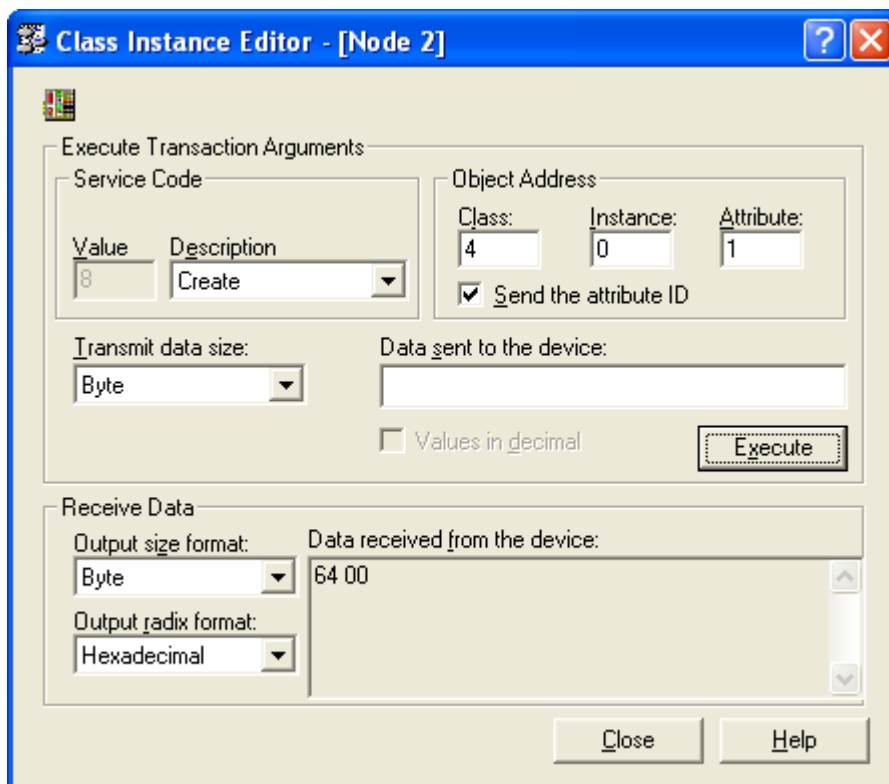
Figure 49: Creating an Instance for Dynamic Assembly

31. If an error occurs, an error message appears: In this case, check the entries
    for Class, Instance and Attribute, the DeviceNet™ connection and
    configuration.

32. Select the "Set Single Attribute" service in the "Description" dialog box and
    enter the following values in the dialog boxes for the "Object Address":

    • "Class"      → "4"
    • "Instance"   → "64" ($64_{hex} = 100_{dec}$)
    • "Attribute"  → "2"

33. Click in the "Data sent to the device" dialog box and enter the following
    values in hexadecimal notation:

    10 00 06 00 20 A6 24 01 30 01 10 00 06 00 20 A6 24 02 30 01

    The path is written with:

0x20 CC (Class) 0x24 II (Instance) 0x30 AA (Attribute)
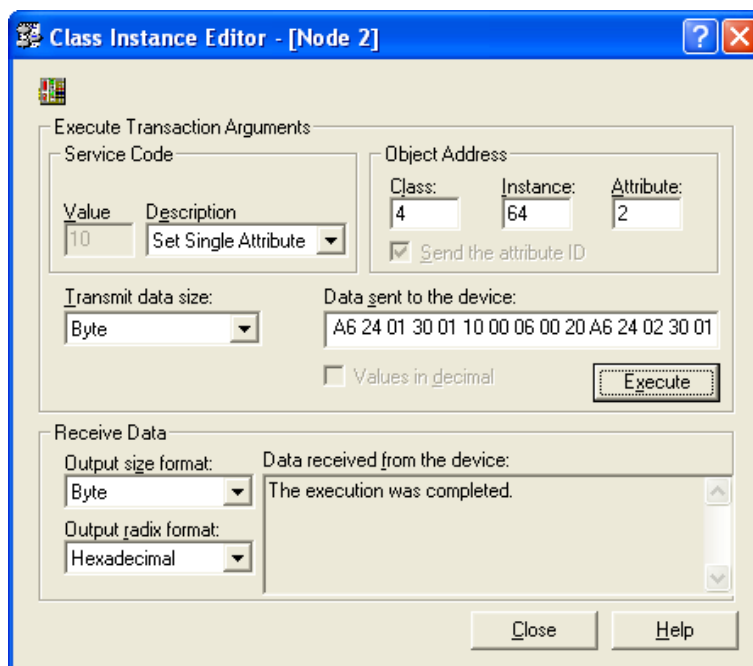


Figure 50: Entering Data to Send

34.    Click the **[Execute]** button to set the mapping.

       If the mapping was successful, the fieldbus node sends confirmation of
       execution.
       If an error occurs, an error message appears.

35.    If a communication or response error occurs, check the DeviceNet™
       connection and instance settings.

36.    Click the **[Close]** button.

The dialog window closes.

37.    To parameterize the fieldbus controller, double-click the fieldbus node 750-
       806 icon.

38.    Select the "Parameters" tab and "All" in the "Groups" dialog box.
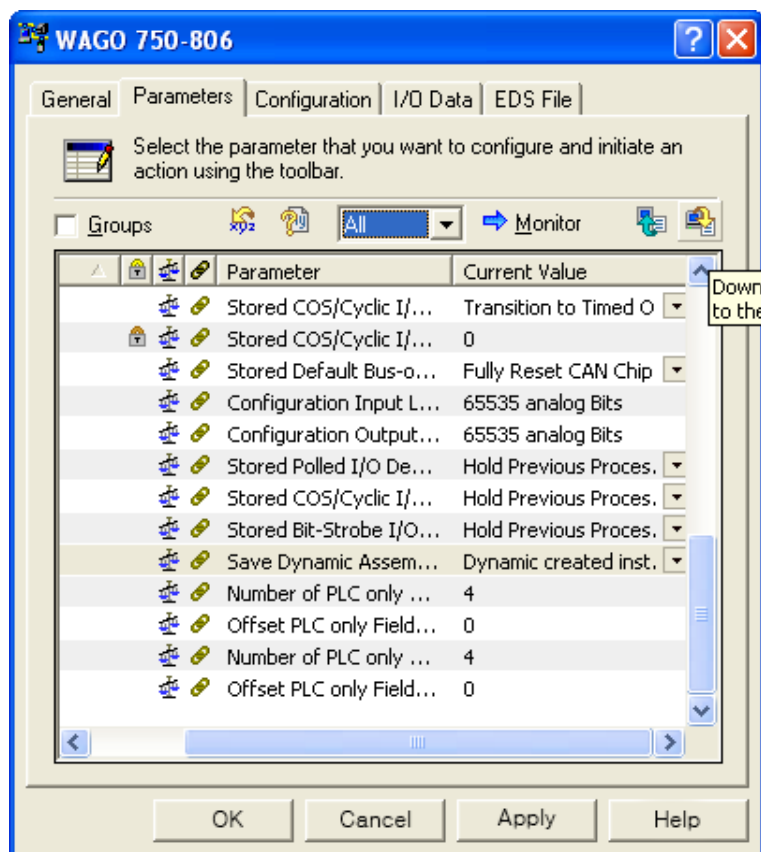
Figure 51: Parameterization

39. Scroll down to the addresses ID#13 and #ID14.

   ID#13 is a pointer to the inputs (default = "4").

   The parameter changes when inputs are mapped for the master. Because inputs can only be read and not written, this is not required.

   ID#14 is a pointer to the outputs (default = "1").

   This parameter changes to point to dynamic mapping of outputs that are mapped in the "Dynamic Assembly Instance" $100_{dec}$. ($0x64_{hex}$).

40. Leave the value for ID#13 at the default value of "4".

   For ID#14, enter 25604 decimal to aim the pointer on the "Dynamic Assembly Output" mapping.

   The value "25604" corresponds to hexadecimal notation "0x6404".
   04 (low byte) = Class type
   64 (high byte) = $100_{dec}$ instance number

41. Change the value for ID#39. Select "Dynamic created instances are stored in non-volatile memory" to make sure the configuration for the "Dynamic Assembly" if the controller loses power.

42. To apply the parameters to the fieldbus controller, select the parameter "All Values" on the "Parameters" tab in the dialog box on the right.

43.    Then click the "Download parameters to the device" icon located at the far right of the dialog box.

44.    Click the **[OK]** button to confirm the setting.

The dialog window closes.

45.    Then switch the fieldbus controller power supply of and on again.

The fieldbus node is now ready for network communication.

# 9 Programming the PFC Using WAGO-I/O-*PRO*

Using IEC 61131-3 programming, the Fieldbus Controller DeviceNet$^{TM}$ 750-806 can also utilize the function of a PLC in addition to the functions of a fieldbus coupler. Creation of an application program in line with IEC 61131-3 is performed using the programming tool WAGO-I/O-*PRO*.

## Information

**Compatibility with IEC-61131-3 programming software!**
The compatibility between your fieldbus controller and the IEC 61131-3 programming software used depends on the fieldbus controller's firmware version and the version of the programming software.
A list of recommended combinations can be found at www.wago.com.
Use the search function (search for "Compatibility Notes").

## Note

**Log in with the administrator password to program the fieldbus controller!**
If you have activated password protection for port 2455 on the "Security" page of the WBM, you have to log into WAGO-I/O-*PRO* in the menu **Online** > **Log In** to obtain programming access to the fieldbus controller (default password "wago").

Figure 52: Logging in for Programming Access

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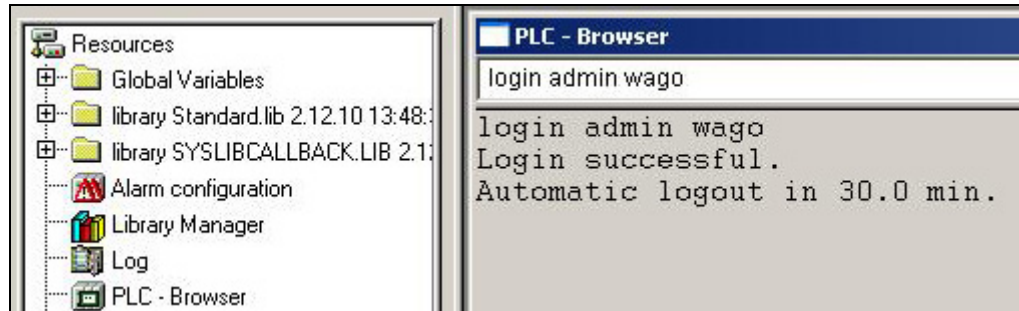


Figure 53: Logging Into the PLC Browser

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

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

> **Note**
>
> **One WAGO-I/O-*PRO*-/(CODESYS)-Instance per traget system!**
> Note that a simultaneous connection of multiple WAGO-I/O-*PRO*/(CODESYS) Instances on one target system is not possible.

> **Note**
>
> **Name Conventions for WAGO-I/O-*PRO/*(CODESYS) Projects!**
> Note that you do not use special characters for the name of your WAGO-I/O-*PRO/*(CODESYS) project and limit the name to a maximum of 8 characters.
> This will ensure that not always, in case of the online change function is activated simultaneously, for each online change event a new TxT file is created, which contains the paths and the project ID, and that for this additional memory is consumed. With proper choice of the file name, the TxT file is only overwritten each time and does not consume additional memory space.

## Information

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

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

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

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

| Target Settings | |
|---|---|
| Configuration: | None ▼ |
| | None |
| | WAGO_750-806 |

Figure 54: Dialog Window for Target System Settings

3.    Select the Fieldbus Controller DeviceNet<sup>TM</sup> 750-806 by entering
      **WAGO_750-806** and then click [**OK**].

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

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

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

## 9.1    Configuring the Fieldbus Controller using the I/O Configurator

The I/O Configurator is a plug-in integrated into WAGO-I/O-*PRO* used to determine addresses for I/O modules at a fieldbus controller.

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

2.    To start the I/O Configurator, double-click in the tree structure on **Control system configuration**.

3.    Expand the branch **Hardware configuration** in the tree structure.

4.    Right-click on the entry **K-Bus** and then select **Edit** in the context menu.

5.    In the "Configuration" window that then opens, click on **Add** to open the module selection window.

6.    Select the I/O module you wish to add from the module catalog and attach it to the end of the internal data bus structure by clicking on **[>>]** and **OK**.

7.    Position all of the required I/O modules in their correct order until this arrangement matches the configuration of the physical node.

Arrange the tree structure in the hardware configuration in the same manner. Include all I/O modules which supply or receive data.

If you access your fieldbus controller online, you can use the **[Start WAGO-I/O-CHECK and scan]** button in the "Configuration" window to read in the physically linked fieldbus controllers with the series-connected I/O modules and display all of the components.

### Note

→ **The internal data bus structure in the WAGO I/O Configurator must match the physical node structure!**

The number of I/O modules that send or receive data must correspond to the existing hardware (except for supply modules, copying modules or end modules, for example). For the number of input and output bits or bytes of the individual I/O modules please refer to their corresponding descriptions.

*Information*

**Additional information**
To open the data sheet for an I/O module, click in the "Configuration" window on the corresponding I/O module and then click the **[Data sheet]** button. The data sheet is then shown in a separate window.
All current data sheets are available on our website http://www.wago.com under Documentation.

8.      Click [**OK]** to accept the node configuration and close the dialog window.

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

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

*Information*

**Additional Information**
For a detailed description of using the software WAGO-I/O-*PRO* and the I/O Configurator, refer to the online Help function for WAGO-I/O-*PRO*.

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

Various libraries are available in WAGO-I/O-*PRO* for different IEC-61131-3 programming tasks. These contain function blocks for universal use that can streamline program creation.

After incorporating the libraries, you can access their function blocks, functions and data types to use just like ones you have defined yourself.

The "standard.lib" library is available by default.
For DeviceNet™-specific projects with WAGO-I/O-*PRO*, the "DevNet. Lib" library is available. This library adds master functionality to the fieldbus controller, so that it can be programmed as the DevivceNet™ master in the network.

**Information**

**Additional Information**
All of these libraries are on the installation CD for the
WAGO-I/O-*PRO* software, or at our website http://www.wago.com.

## 9.3      General Information about IEC Tasks

Please note the following information when programming your IEC tasks:

*Note*

**Use different priorities for IEC tasks!**
Give IEC tasks (in WAGO-I/O-*PRO* under the **Resources** tab > **Task Configuration**) different priorities to prevent an error from occurring when compiling the user program.

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

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

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

Example:      $1^{st}$  Task 4 ms → Waiting period 2 ms
              $2^{nd}$ Task 2 ms → Waiting period 1 ms

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

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

The following applies to cyclic tasks with watchdog activated:

---

**Note**

→  **Reference for Watchdog Settings!**
For each tasks created, a watchdog can be enabled that monitors the execution time of a task.
If the task runtime exceeds the specified watchdog time (e.g., t#200 ms), then the watchdog event has occurred.
The runtime system stops the IEC program and reports an error.
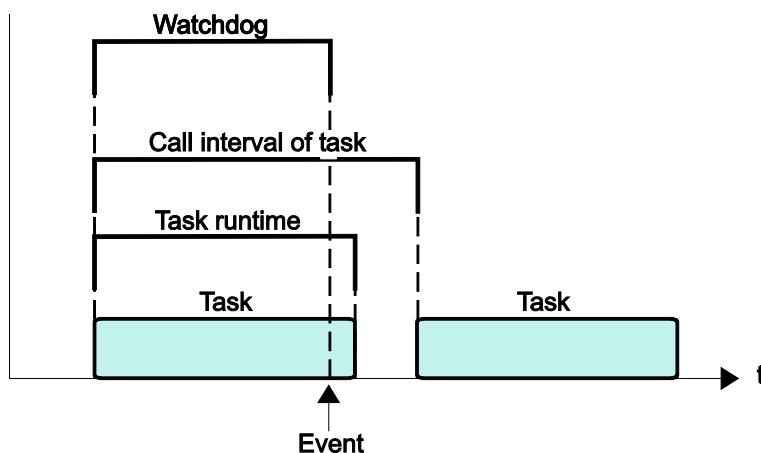
Figure 55: Watchdog Runtime is Less Than the Task Runtime

If the watchdog time set is greater than the call interval of the task, then the watchdog is restarted for each task call interval.
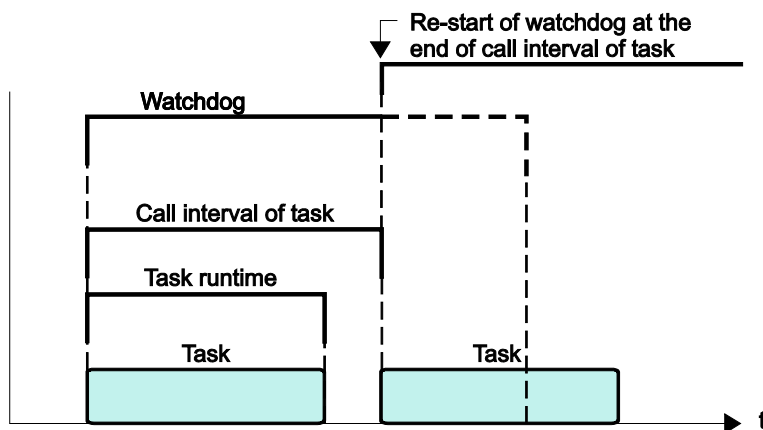
Figure 56: Watchdog Runtime Is Greater Than the Task Call Interval

### 9.3.1    IEC Task Sequence

1.    Determine the system time (tStart).

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

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

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

5.    Writing of the outputs to the process image.

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

### 9.3.2    Overview of Most Important Task Priorities

Table 40: Task Processing

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

**Internal bus task/fieldbus task (internal)**
The internal bus task matches the process image to the input/output data of the modules in defined cycles. The fieldbus tasks are performed as triggered by events and only require computing time when communication is performed via fieldbus (MODBUS).

**Normal task (IEC-Task priorities 1-10 that can be set in CODESYS)**
IEC tasks with this priority may be interrupted by the internal bus tasks. Therefore, configuration for the connected modules and communication via fieldbus with the watchdog activated for the task call interval must be taken into account here.

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

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

**Information**

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

## 9.4    System Events

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

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

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

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

## 9.4.1 Enabling/Disabling System Events

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

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

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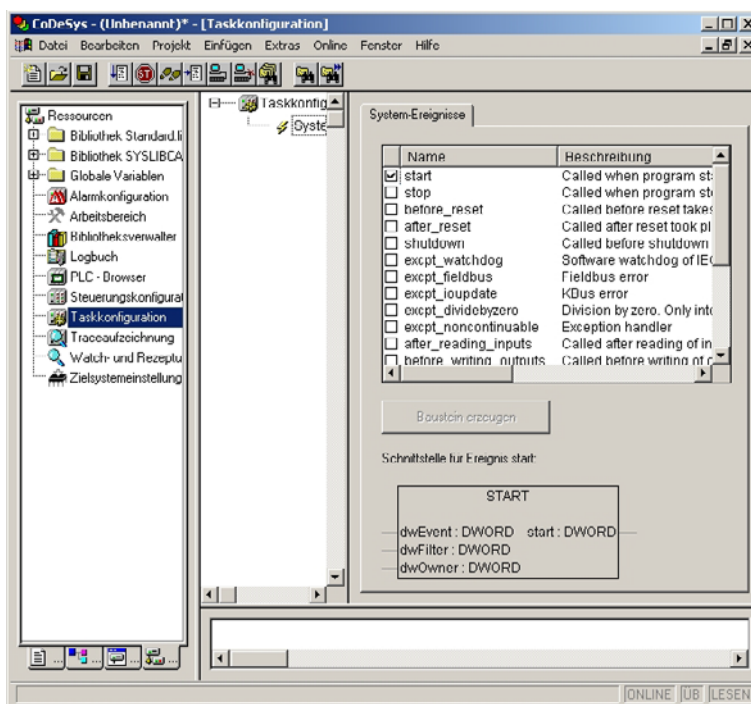


Figure 57: Enabling/Disabling System Events

→ **Note**

**Only use global variables and functions (Fun)!**
Note that the fetch of function blocks (FB) such as Timer modules , or the fetch of blocking functions such as file operations can lead to a malfunction of the system. Therefore use in system events exclusively Global variables and functions (Fun).

ℹ **Information**

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

## 9.5 Transfer the IEC program to the Fieldbus controller

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

- Direct transfer via serial port
- Transfer via fieldbus

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

**Note**

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

**Note**

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

**Note**

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

**Note**

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

**Information**

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

## 9.5.1    Transfer via Serial Service Port

**Note**

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

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

**NOTICE**

**Do not connect Communication Cable when energized!**
To prevent damage to the communications interface, do not connect or disconnect Communication Cable 750-920 respectively 750-923 when energized!
The fieldbus controller must be de-energized!

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

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

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

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

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

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

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

WAGO®
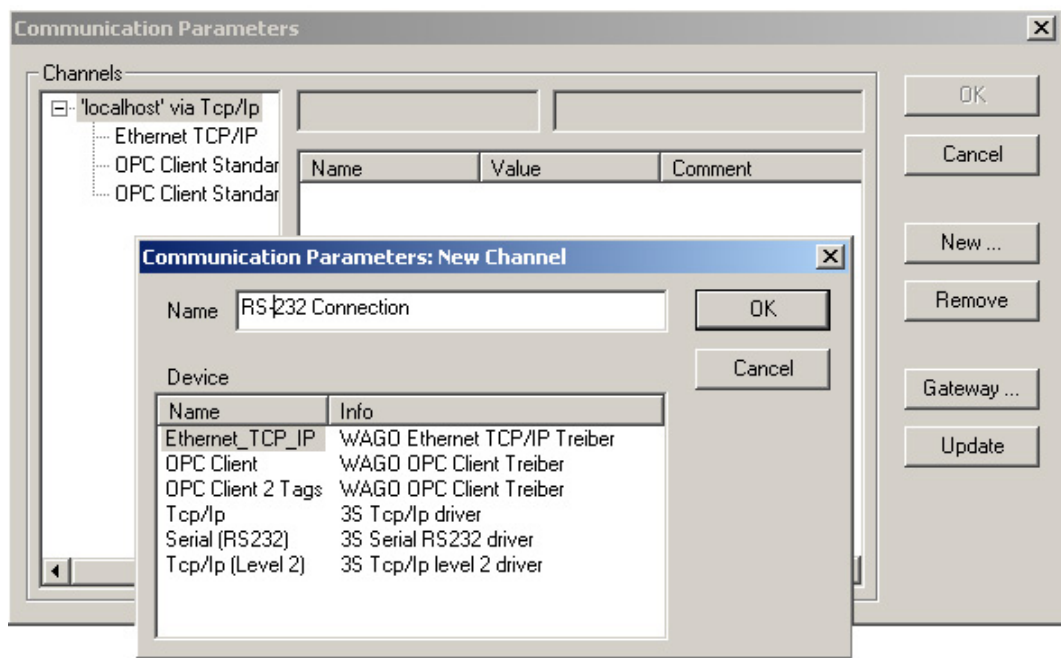
Figure 58: Dialog Window "Communication Parameters"

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

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

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

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

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

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

## Note

**Log in with the administrator password to program the fieldbus controller!**
If you have activated password protection for port 2455 on the "Security" page of
the WBM, you have to log into WAGO-I/O-*PRO* in the menu **Online** > **Log In** to
obtain programming access to the fieldbus controller (default password "wago").

Figure 59: Logging in for Programming Access

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

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

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

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

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

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

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

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

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

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

## 9.5.2    Transferring an Application via Fieldbus

The fieldbus cable physically connects the PC and the fieldbus controller. It is necessary to have a suitable communication driver for data transmission. Enter the driver and its parameters in WAGO-I/O-*PRO* in the "Communication parameters" dialog window:

---

**Note**

**Note the position of the mode selector switch!**
The mode selector switch located behind the cover cap of the fieldbus controller next to the service interface must be in the center or top position.

---

**Note**

**View WAGO-I/O-*PRO* as a subscriber in terms of UCMM!**
Transfer via the fieldbus is supported by UCMM. WAGO-I/O-*PRO* counts for downloading the PFC program as a subscriber.

---

1.  Launch the WAGO-I/O-*PRO* software under **Start → Programs → WAGO Software → WAGO-I/O-*PRO***.

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

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

3.  Click **New...** to establish a new connection and assign a name.

4.  Select the required driver on the right side of the dialog to configure the connection between the PC and fieldbus controller.

5.  Use the "Hilscher PA Interface standard" driver.

6.  Change the entries as needed in the center dialog window.

7.  Click **[OK]** to confirm.

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

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

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

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

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

---

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

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

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

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

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

# 9.6    Special DeviceNet$^{TM}$ Characteristics of the Fieldbus Controller

## 9.6.1    Establishing a Connection via the UCMM Port

The DeviceNet$^{TM}$ fieldbus controller supports dynamic connection setup from the "Unconnected Message Manager Port" (UCMM port).

The fieldbus controller can be set up with 5 explicit and 5 dynamic incoming and outgoing connections at the same time, i.e., 5 subscribers can be connected at the same time.

## 9.6.2    Offline Connection Set

Using the "Offline Connection Set", the fieldbus node can be addressed across the network if switched off due to a duplicate DeviceNet$^{TM}$ MAD ID and it is in the "Communication Fault" state.

After addressing the fieldbus node, the DeviceNet$^{TM}$ MAC ID of the fieldbus controller can be changed from the software.

## 9.6.3    DeviceNet Shutdown

"DeviceNet Shutdown" allows the fieldbus node to log out of a controller when the node is switched off due to internal error. This function can be used in DeviceNet$^{TM}$ networks specifically where very high safety requirements apply, e.g., in chemical and semiconductor manufacturing.

## 9.6.4    Dynamic Assembly

An "Assembly Object" is used to combine data (e.g., input/output data) into blocks to send them in a single message.

The static assembly can be used to access fixed pre-programmed assembly instances in the fieldbus controller. In contrast, the dynamic assembly makes it possible to set up assembly instances in which process data can be compiled from various applications as required.

In addition to transmitting input/output data, the dynamic assembly can also be used for targeted selection of data that should be transmitted explicitly or not transmitted explicitly via the fieldbus.

**Note**

**Do not apply physical outputs in mapping!**
If setting physical outputs from the PFC, either instance 11 of the static assembly or the dynamic assembly must be used. The physical outputs may not be applied in the mapping. This prevents output data from being transmitted via the fieldbus and being overwritten by it in the short term.

**Information**

**Additional Information**
More details are available in section "Fieldbus Communication" > "Object Classes" > "Assembly Object (0x04)".

## 9.6.5    Change MAC ID by SW

Software (e.g., RSNetWorx) can be used to change the MAC ID of the DeviceNet™ fieldbus controller. The node address is stored in a non-volatile memory. If the address set on the DIP switch differs from the address set by software via the network, the color of the I/O LED changes to orange. To reset the address assigned by software, invalid address 64 is entered in Class 3, Instance 1, Attribute 1. The controller then has the MAD ID set on the DIP switch.

## 9.6.6    Heartbeat

The Heartbeat function allows a node to send a so-called Heartbeat message and to make known its ability to communication to all network subscribers.

If a responsible "Heartbeat Consumer" does not receive any message in a defined time (Heartbeat Consuming Time), it is registered as a heartbeat error. The relationships between the producer and consumer of a heartbeat message can be configured by entries in the object directory, i.e., the time between two heartbeat messages can be entered in Class 0x01, Instance 1, Attribute ID 10 (0x0A).

## 9.6.7    Bit-Strobe

The "Bit-Strobe I/O Connection" is always a 1 to n multicast connection. In other words, a master can reach all slaves that support the Bit-Strobe command. Transmission is simultaneous. This allows for slave synchronization.

The length of this master message is limited to 8 bytes. Each node address in the network is assigned a bit within the 8 data bytes. The reaction of the slave, whose bit is set, is specific to the application. The slave responses must be defined and known to the PLC. Each slave can return 8 bytes of data as a response. The order of responses depends on the response time of the individual slaves and the respective node address. If all slaves response to the Bit-Strobe command at the same time, the order of sending on the CAN bus would be determined by the node address (bit arbitration).

**Information**

**Additional Information**
More details are available in section "Fieldbus Communication" > "DeviceNet$^{TM}$" > "Object Model".

# 10   Diagnostics

## 10.1   LED Signaling

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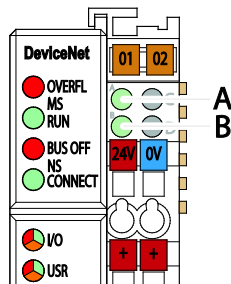


Figure 60: Display Elements

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

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

Table 41: LED Assignment for Diagnostics

| Diagnostics area | LEDs |
|---|---|
| Fieldbus status | • MS<br>  -OVERFL<br>  -RUN<br>• NS<br>  -BUS OFF<br>  -CONNECT |
| Node status | • I/O<br>• USR |
| Status Supply Voltage | • A  (system supply)<br>• B  (field supply) |

## 10.1.1    Evaluating the Fieldbus Status

Communication via the fieldbus is by the top LED group. The two "MS" ("Module Status") and "NS" ("Network Status") LEDs are used for the status of the system and fieldbus connections.

Table 42: Diagnostics – Module Status (MS)

| Module Status (MS) | | | |
|---|---|---|---|
| **OVERFL (red)** | **RUN (green)** | **Device status by DeviceNet™** | **Explanation** |
| off | off | "no power" | The device has no power. |
| off | on | "device operational" | The device is working properly. |
| off | flashes | "device in standby" | The device must still be configured or is partially configured. |
| flashes | off | "minor fault" | There is a minor error. A diagnosis is available. |
| on | off | "unrecoverable fault" | The device is defective and requires service or must be replaced. |
| flashes | flashes | "device self testing" | The device is performing a self-test. |

Table 43: Diagnostics – Network Status (NS)

| Network Status (NS) | | | |
|---|---|---|---|
| **BUS OFF (red)** | **CONNECT (green)** | **Device status by DeviceNet™** | **Explanation** |
| off | off | No power or not online | The device is has no power (fieldbus power supply), the DeviceNet™ cable is not attached and the "Duplicate MAC ID detection" has not been completed. |
| off | flashes | Online but not connected | The device is working properly on the fieldbus. However, but yet integrated by a scanner. |
| off | on | Online and connected | The device is working properly on the fieldbus. There is at least one connection to another device. |
| flashes | off | Error (time out) | There is a minor error (e.g., EPR at a Poll connection not equal to 0, slave is no longer polled cyclically). |
| on | off | Critical connection error | The device has detected an error "Duplicated MAC ID Check"). No functions can be executed on the network. |

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

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

Table 44: Node Status Diagnostics – Solution in Event of Error

| LED Status | Meaning | Solution |
|---|---|---|
| **I/O** | | |
| green | The fieldbus node is operating correctly. | Normal operation. |
| orange flashing | Start of the firmware. 1 … 2 seconds of rapid flashing indicate start-up. | - |
| red | Fieldbus coupler/controller hardware defect | Replace the fieldbus coupler/controller. |
| red flashing | Flashing with approx.. 10 Hz indicates the initialization of the internal bus or of a internal bus error. | Note the following flashing sequence. |
| red cyclical flashing | Up to three successive flashing sequences indicate internal data bus errors. There are short intervals between the sequences. | Evaluate the flashing sequences based on the following blink code table. The blinking indicates an error message comprised of an error code and error argument. |
| off | No data cycle on the internal bus. | The fieldbus coupler/controller supply is off. |

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

Then the bus is initialized. This is indicated by flashing red at 10 Hz for 1 … 2 seconds.

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

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

After elimination of the error, restart the node by turning the power supply of the device off and on again.
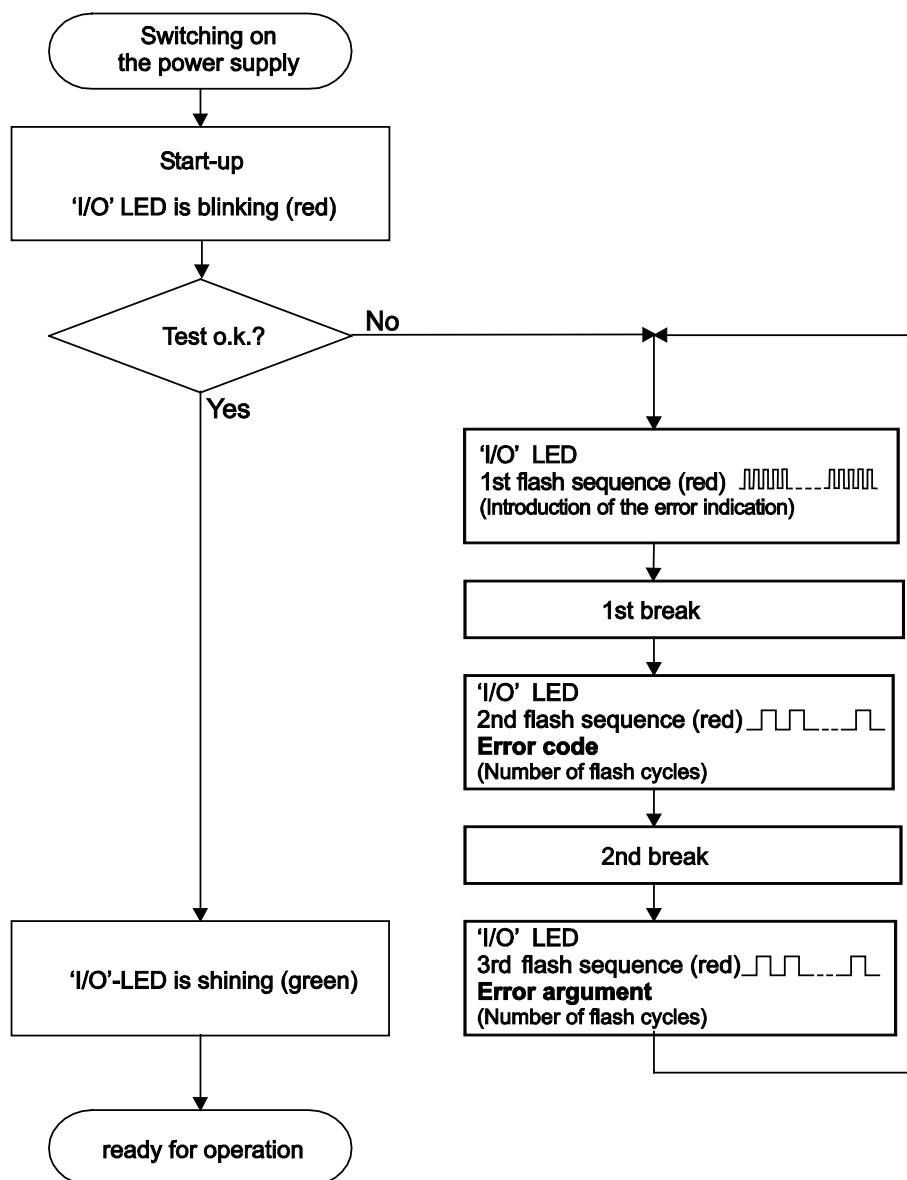
Figure 61: Node Status – I/O LED Signaling
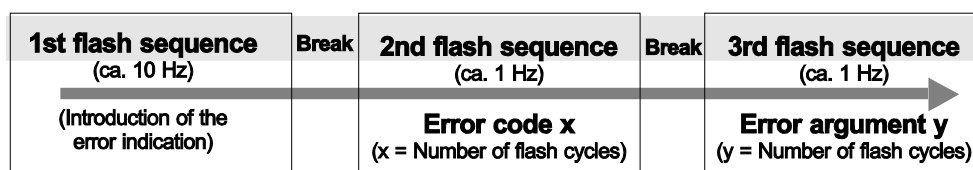


Figure 62: Error Message Coding

## Example of a module error:

• The I/O LED starts the error display with the first flashing sequence (approx. 10 Hz).

• After the first break, the second flashing sequence starts (approx. 1 Hz): The I/O LED blinks four times.
  Error code 4 indicates "data error internal data bus".

- After the second break, the third flashing sequence starts (approx. 1 Hz):
The I/O LED blinks twelve times.
Error argument 12 means that the internal data bus is interrupted behind the
twelfth I/O module.

The thirteenth I/O module is either defective or has been pulled out of the
assembly.

Table 45: Blink Code- Table for the I/O LED Signaling, Error Code 1

| Error code 1: "Hardware and configuration error" | | |
|---|---|---|
| **Error Argument** | **Error Description** | **Solution** |
| **-** | Invalid check sum in the parameter area of the fieldbus controller. | 1. Turn off the power supply for the node.<br>2. Replace the fieldbus controller.<br>3. Turn the power supply on again. |
| **1** | Overflow of the internal buffer memory for the attached I/O modules. | 1. Turn off the power for the node.<br>2. Reduce the number of I/O modules.<br>3. Turn the power supply on again.<br>4. If the error persists, replace the fieldbus controller. |
| **2** | I/O module(s) with unknown data type | 1. Determine the faulty I/O module by first turning off the power supply.<br>2. Plug the end module into the middle of the node.<br>3. Turn the power supply on again.<br>4. --- LED continues to flash? ---<br>Turn off the power supply and plug the end module into the middle of the first half of the node (toward the fieldbus controller).<br>--- LED not flashing? ---<br>Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller).<br>5. Turn the power supply on again.<br>6. Repeat the procedure described in step 4 while halving the step size until the faulty I/O module is detected.<br>7. Replace the faulty I/O module.<br>8. Inquire about a firmware update for the fieldbus controller. |
| **3** | Unknown module type of the Flash program memory | 1. Turn off the power supply for the node.<br>2. Replace the fieldbus controller.<br>3. Turn the power supply on again. |
| **4** | Fault when writing in the Flash program memory. | 1. Turn off the power supply for the node.<br>2. Replace the fieldbus controller.<br>3. Turn the power supply on again. |
| **5** | Fault when deleting the Flash memory. | 1. Turn off the power supply for the node.<br>2. Replace the fieldbus controller.<br>3. Turn the power supply on again. |

Table 45: Blink Code- Table for the I/O LED Signaling, Error Code 1

| **Error code 1: "Hardware and configuration error"** | | |
|---|---|---|
| **Error Argument** | **Error Description** | **Solution** |
| **6** | The I/O module configuration after AUTORESET differs from the configuration determined the last time the fieldbus controller was powered up. | 1.   Restart the fieldbus controller by turning the power supply off and on. |
| **7** | Fault when writing in the serial EEPROM. | 1.   Turn off the power supply for the node.<br>2.   Replace the fieldbus controller.<br>3.   Turn the power supply on again. |
| **8** | Invalid hardware-firmware combination. | 1.   Turn off the power supply for the node.<br>2.   Replace the fieldbus controller.<br>3.   Turn the power supply on again. |
| **9** | Invalid check sum in the serial EEPROM. | 1.   Turn off the power supply for the node.<br>2.   Replace the fieldbus controller.<br>3.   Turn the power supply on again. |
| **10** | Serial EEPROM initialization error | 1.   Turn off the power supply for the node.<br>2.   Replace the fieldbus controller.<br>3.   Turn the power supply on again. |
| **11** | Fault when reading in the serial EEPROM. | 1.   Turn off the power supply for the node.<br>2.   Replace the fieldbus controller.<br>3.   Turn the power supply on again. |
| **12** | Timeout during access on the serial EEPROM | 1.   Turn off the power supply for the node.<br>2.   Replace the fieldbus controller.<br>3.   Turn the power supply on again. |
| **14** | Maximum number of gateway or mailbox modules exceeded | 1.   Turn off the power for the node.<br>2.   Reduce the number of corresponding modules to a valid number.<br>3.   Turn the power supply on again. |

Table 46: Blink Code Table for the I/O LED Signaling, Error Code 2

| Error code 2: "Exceeded Process Image" | | |
|---|---|---|
| **Error Argument** | **Error Description** | **Solution** |
| **1** | Not used | - |
| **2** | Process image is too large. | 1. Turn off the power supply of the node. 2. Reduce number of I/O modules. 3. Turn the power supply on. |

Table 47: 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. <br><br> - Are all modules connected correctly or are there any 750-613 Modules in the node? - 1. Determine the faulty I/O module by turning off the power supply. 2. Plug the end module into the middle of the node. 3. Turn the power supply on again. 4. - LED continues to flash? - Turn off the power supply and plug the end module into the middle of the first half of the node (toward the fieldbus controller). - LED not flashing? - Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller). 5. Turn the power supply on again. 6. Repeat the procedure described in step 4 while halving the step size until the faulty I/O module is detected. 7. Replace the faulty I/O module. 8. If there is only one I/O module on the fieldbus controller and the LED is flashing, either the I/O module or fieldbus controller is defective. 9. Replace the defective component. |

Table 48: Blink Code Table for the I/O LED Signaling, Error Code 4

| Error code 4: "Physical error, internal bus" | | |
| --- | --- | --- |
| Error Argument | Error Description | Solution |
| - | Internal bus data transmission error or interruption of the internal data bus at the fieldbus controller | 1. Turn off the power supply to the node.<br>2. Plug in an end module behind the fieldbus controller.<br>3. Turn the power supply on.<br>4. Observe the error argument signaled.<br><br>- Is no error argument indicated by the I/O LED? -<br>5. Replace the fieldbus controller.<br><br>- Is an error argument indicated by the I/O LED? -<br>5. Identify the faulty I/O module by turning off the power supply.<br>6. Plug the end module into the middle of the node.<br>7. Turn the power supply on again.<br>8. - LED continues to flash? -<br>Turn off the power and plug the end module into the middle of the first half of the node (toward the fieldbus controller).<br>- LED not flashing? -<br>Turn off the power and plug the end module into the middle of the second half of the node (away from the fieldbus controller).<br>9. Turn the power supply on again.<br>10. Repeat the procedure described in step 6 while halving the step size until the faulty I/O module is detected.<br>11. Replace the faulty I/O module.<br>12. If there is only one I/O module on the fieldbus controller and the LED is flashing, either the I/O module or fieldbus controller is defective.<br>13. Replace the defective component. |
| n* | Interruption of the internal data bus behind the nth bus module with process data, the maximum supported number is reached, the following modules are no longer supported. | 1. Turn off the power supply of the node.<br>2. Reduce number of I/O modules.<br>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 49: 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.<br>2. Replace the (n+1) I/O module containing process data.<br>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 50: Blink Ccode Table for the I/O LED Signaling, Error Code 6 … 8

| Error code 6 … 8:   -not used- | | |
|---|---|---|
| **Error Argument** | **Error Description** | **Solution** |
| **-** | Not used | |

Table 51: Blink Code Table for the I/O LED Signaling, Error Code 9

| Error code 9: "CPU Trap error" | | |
|---|---|---|
| **Error Argument** | **Error Description** | **Solution** |
| **1** | Illegal Opcode | Fault in the program sequence.<br>1.   Please contact the I/O Support. |
| **2** | Stack overflow | |
| **3** | Stack underflow | |
| **4** | NMI | |

Table 52: Blink Code Table for the 'I/O' LED Signaling, Error Code 10

| Error code 10: "PLC program fault" | | |
|---|---|---|
| **Error Argument** | **Error Description** | **Solution** |
| **1** | Invalid Offset address for digital inputs | 1.   Correct the Offset address in the appropriate function block. |
| **2** | Invalid Offset address for digital outputs | 1.   Correct the Offset address in the appropriate function block. |

Table 53: Blink Code Table for the I/O LED Signaling, Error Code 11

| Error code 11: "Error in I/O modules with mailbox functionality" | | |
|---|---|---|
| Error Argument | Error Description | Remedy |
| 1 | Too many I/O modules with mailbox functionality are plugged in | 1.   Reduce the number of mailbox modules. |
| | **Please note the firmware version.** The fieldbus controller with firmware version 11 supports 3 I/O modules with mailbox functionality. Firmware version 12 supports 8 I/O modules with mailbox functionality. **Note process image size limit!** Please note the process image size limit when configuring a node with analog input/output modules and I/O modules that have mailbox functionality. Depending on the overall configuration of all I/O modules of a node, the maximum number of object directory entries may be exceeded in some cases. | |
| 2 | Maximum mailbox size exceeded | 1.   Reduce the size of the mailbox. |
| 3 | Maximum PA size exceeded due to connected I/O modules with mailbox functionality | 1.   Reduce the data width of I/O modules with mailbox functionality. |

### 10.1.2.1   USR LED

The bottom indicator LED ("USR") is provided for visual output of information.

Control of the LED from the application program is conducted using the functions from the WAGO-I/O-*PRO* library "Visual.lib".

## 10.1.3   Evaluating Power Supply Status

The power supply unit of the device has two green LEDs that indicate the
status of the power supplies.
LED "A" indicates the 24 V supply of the coupler.
LED "B" or "C" reports the power available on the power jumper contacts
for field side power.

Table 54: Power Supply Status Diagnostics – Solution in Event of Error

| LED Status | Meaning | Solution |
|---|---|---|
| **A** | | |
| Green | Operating voltage for the system is available. | - |
| Off | No power is available for the system | Check the power supply for the system (24 V and 0 V). |
| **B or C** | | |
| Green | The operating voltage for power jumper contacts is available. | - |
| Off | No operating voltage is available for the power jumper contacts. | Check the power supply for the power jumper contacts (24 V and 0 V). |

# 11      Fieldbus Communication

## 11.1     DeviceNet<sup>TM</sup>

DeviceNet<sup>TM</sup> is a network concept on the device level based on the serial bus
system "Controller Area Network" (CAN). It is particularly characterized by easy
addition and removal of devices during operation. The range of devices spans
from simple light barriers to complex engine control units. DeviceNet<sup>TM</sup> is
primarily used in industrial automation and robot controllers.

The physical data link layer is defined in the CAN specification. The telegram
structure is described, but nothing is said about the application layer. DeviceNet<sup>TM</sup>
is implemented here. It describes the significance of the transmitted data defined
in the application layer. The "Open DeviceNet<sup>TM</sup> Vendor Association" (ODVA) is
the user organization for DeviceNet<sup>TM</sup>. In a specification, the ODVA devices
DeviceNet<sup>TM</sup> as a uniform application layer and specifies technical and functional
characteristics for device linking.

Up to 64 fieldbus nodes can be operated in one DeviceNet<sup>TM</sup> network. Network
reach depends on the selected baud rate (125 kBaud, 250 kBaud or 500 kBaud).
Contrary to other fieldbus systems, in CAN the modules connected to the bus are
not addressed, but the messages are identified.

The devices are allowed to send messages whenever the bus is available. Each bus
node decides by itself when it wants to send data or prompts other bus nodes to
send data. Thus, communication without the bus master module is possible. Bus
conflicts are solved by assigning messages a specific priority. This priority is
defined by the CAN identifier, "Connection ID" at DeviceNet<sup>TM</sup>. The smaller the
identifier is, the higher the priority.

A general difference between high-priority process messages (I/O messages) and
low-priority management messages (explicit messages) is done before. Messages
with a data length of more than 8 bytes can be fragmented.

Communication with DeviceNet<sup>TM</sup> is always connection based. All data and
functions of a device are described by means of an object model. Therefore, for a
message exchange directly after switching on a device, the connections to the
desired subscriber have to be established first and communication objects be
created or allocated. Message distribution is according to the broadcast system,
data exchange according to the producer consumer model.

A transmitting DeviceNet<sup>TM</sup> node produces data that is either consumed via a
point-to-point connection (1 to 1) by one receiving node, or via a multicast
connection (1 to n) by several receiving nodes.

**Information**

**Additional Information**
The "Open DeviceNet™ Vendor Association" (ODVA) makes more information available on the Internet at: http://www.odva.org.

**Information**

**Additional Information**
"CAN in Automation" (CiA) makes documentation about CAN networks available on the Internet at: http://www.can-cia.de.

### 11.1.1    Network Structure

#### 11.1.1.1    Transfer Media

A bus medium forms the basis for the physical implementation of a network with DeviceNet™.

According to the cable specification, a double 2-conductor twisted pair cable (twisted pair, shielded cable) is recommended ad the medium. It consists of two shielded twisted-pair cables with a wire in the middle of the cable. Additional shielding runs on the outside. The blue and white twisted-pair cable is used for signal transmission, the black and red one for the power supply.

The DeviceNet™- bus is configured from a remote bus cable as the trunk line and several drop lines.

The DeviceNet™ specification distinguishes between 2 cable types:

- **Thick Cable**
  For the trunk line with maximum 8 A or for networks extending over more than 100 m.
  The trunk line topology is linear, i.e., remote bus cables are not further branched. At each end of the remote bus cable, terminating resisters are required.

- **Thin Cable**
  For drop lines with maximum 3 A or for networks extending less than 100 m.
  One or more nodes can be connected to the drop lines, i.e., branching is permitted here. The length of the individual drop lines is measured from the branching point of the node and can be up to 6 m The entire length of the drop line depends on the baud rate.

**NOTICE**

**Note information about connection data lines!**
Route data lines separately from all high-current cables.

**Information**

**Additional Information**
The detailed specification regarding cable types is available on the Internet at:
http://www.odva.org.

In the following table, the permitted cable length is represented based on the baud rate. A distinction is made between the maximum lengths for transmission with thick and thin cable.

Table 55: Maximum bus lengths dependent on preset baud rate

| Baud rate | Bus length | | | Drop Cable Length | |
|---|---|---|---|---|---|
| | Thick + Thin Cable | Thick Cable only | Thinn Cable only | maximum | cumulated |
| 500 Kbit/s | $L_{Thick} + L_{Thin} \leq 100$ m (328 ft) | 100 m (328 ft) | 100 m (328 ft) | 6 m (19.6 ft) | 39 m (127.9 ft) |
| 250 Kbit/s | $L_{Thick} + 2.5 \cdot L_{Thin} \leq 250$ m (820.2 ft) | 250 m (820.2 ft) | 100 m (328 ft) | 6 m (19.6 ft) | 78 m (255.9 ft) |
| 125 Kbit/s | $L_{Thick} + 5 \cdot L_{Thin} \leq 500$ m (1640.4 ft) | 500 m (1640.4 ft) | 100 m (328 ft) | 6 m (19.6 ft) | 156 m (511.8 ft) |

Specifying maximum cable lengths ensures that communication is possible between two nodes located at maximum distance to each other (worst case).

## 11.1.1.2  Cabling

The connection of a WAGO fieldbus node to the DeviceNet™ bus cable is made by the included 5-pole plug, Series 231 *(MCS)*.
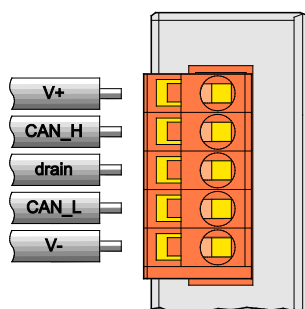


Figure 63: Plug Assignment for the Fieldbus Connection, Series 231 *(MCS)*

For wiring with shielded cable, the connector is assigned the connections V+, V- for the power supply and CAN_High, CAN_Low for data transmission

The 24 V field bus supply is fed by an external fieldbus network power supply.

CAN_High and CAN_Low are two physically different bus signal levels. The cable's shielding is connected to the "drain" connection. This is terminated to PE in devices with 1 MΩ (DIN rail contact). A low-impedance connection of the shielding to PE is possible only from the outside (e.g., by a supply module). The aim is a central PE contact for the entire DeviceNet™ bus cable shield.

## Note

**Use the WAGO Shield Connecting System for optimal shielding!**
For the optimal connection between fieldbus cable shielding and functional ground, WAGO offers a cable shielding system (Series 790).

Each DeviceNet™ node forms the UDiff differential voltage from the CAN_High and CAN_Low as follows: UDiff = UCAN_High - UCAN_Low.

The differential signal transmission offers the benefit of insensitivity to common mode interferences and ground offsets between the nodes.

## Note

**Use the proper terminating resistor for both ends of the bus cable!**
The bus cable must have a terminating resistor of 121 Ω / ±1% / ¼ W at both ends between CAN-High and CAN-Low to prevent reflections and transmission problems.
This is also required for very short cable lengths.

Because the CAN bus can be designed as a 2-wire bus, bus fault management detects a break or short circuit in a line by asymmetric operation.

## Information

**Additional Information**
The "CAN in Automation" (CiA) organization makes specification documents for a CAN network available on the Internet at: http://www.can-cia.de.

### 11.1.1.3  Topology

To build a simple DeviceNet™ network, you need a scanner (PC with DeviceNet™ fieldbus PCB card), a connection cable and a 24 VDC power supply unit in addition to a DeviceNet™ fieldbus node.

The DeviceNet™ network is set up as a line structure (trunk line) with terminating resistors (121 Ohm).
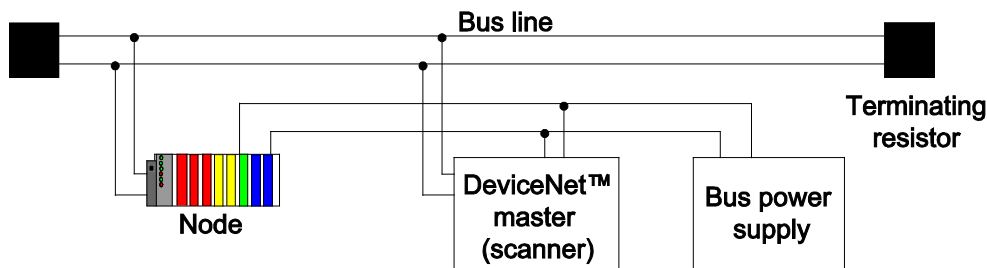
Figure 64: DeviceNet™ Network – Line Structure (Trunk Line) with Terminating Resistors

In systems with more than two stations, all nodes are wired in parallel. The nodes are connected to the fieldbus cable (trunk line) by drop lines. This requires the bus cable to be looped through without disruption. The maximum length of a line branch should not exceed 6 m.

An example of this topology is shown in the following figure:
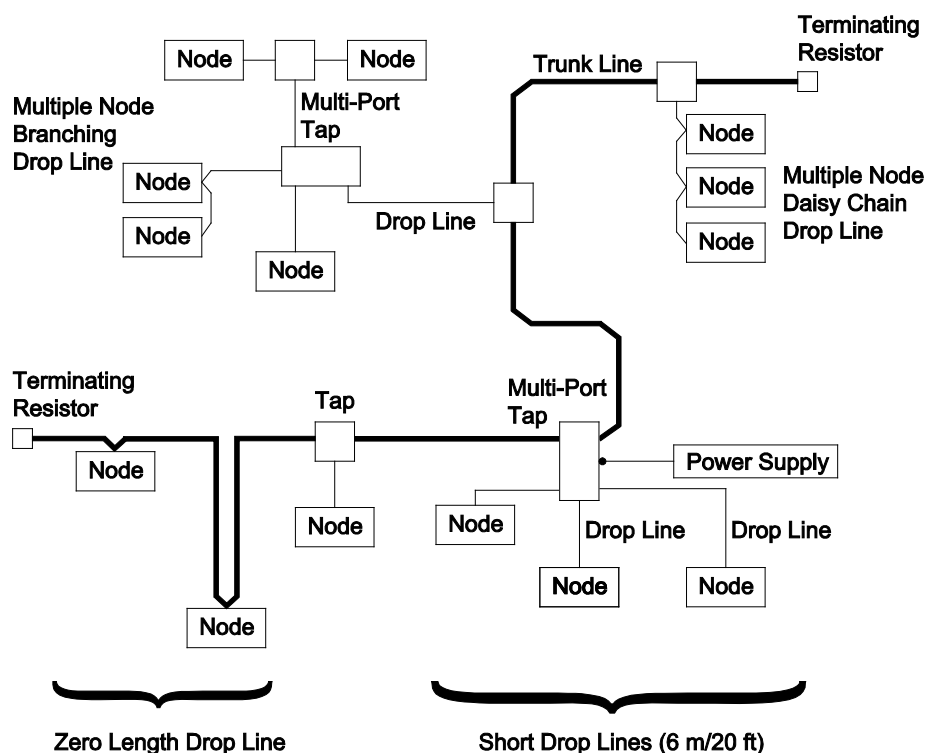


Figure 65: DeviceNet™ Network with Cable Branches

To connect the nodes, a branching unit ("Multi-Port DeviceNet™ Tap") has been developed by WAGO Kontakttechnik GmbH & Co. KG. The unit allows remote bus cables and drop lines to be connected using CAGE CLAMP® technology. A secure and fast, as well as vibration- and corrosion-resistant connection is thereby achieved.

The DeviceNet™ taps are available in 2 versions.

Table 56: DeviceNet™ Tap Versions

| Items | Description |
|---|---|
| 810-900/000-001 | Closed design with connection options for 6 lines. The enclosure provides protection in harsh environments. |
| 810-901/000-001 | Open design for connecting 2 drop lines and 2 remote bus cables. |

All devices in the network communicate at the same baud rate. The bus structure makes it possible to couple and decouple stations or to start up the system step by step in a non-reactive manner.

Later upgrades have no effect on stations that are already in operation. If a device malfunctions or is added to the network, it is automatically detected.

### 11.1.1.4   Network Grounding

The devices can either be powered by the DeviceNet™ bus or have their own power supply. However, the network can only be grounded at one location. The network is preferably grounded in the network center (surge arrestor V- and "drain" shielding for round media) to optimize capacity and to minimize interference.

Not permitted are ground loops via devices that are not disconnected from the power supply. The device must either be insulated or, if that is not possible, the power must be disconnected from the device.

### 11.1.1.5   Interface Modules

In a network, all WAGO DeviceNet™- fieldbus nodes are delivered to operate as slaves in a network. Master operation is performed by a central control system, such as PLC, NC or RC.

> **Note**
>
> **Integrate library for master functionality!**
> The programmable fieldbus controller 750-806 can assume the master operation when extended with the "DevNet.lib" library.

The fieldbus devices are linked via interface modules.

As an interface module, WAGO offers the PC interface cards for DeviceNet™, ISA DeviceNet™ Master 7 kByte (order No. 758-340), PC104 DeviceNet™ Master 7 kByte D-Sub, straight, angled (order No. 758-341) and PCI DeviceNet™ Master 7 kByte (order No. 758-342) from the WAGO-I/O-SYSTEM 758.

Other interface modules for programmable logic controllers are also available from other manufacturers.

## 11.1.2    Network Communication

### 11.1.2.1    Objects, Classes, Instances and Attributes

Protocol processing of DeviceNet<sup>TM</sup> is object oriented. Each node in the network is depicted as a collection of objects. Some related terms are defined below:

- **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 contains related components (objects) of a product organized in instances, e.g., Identity Class, DeviceNet Class.

- **Instance**
  An instance is composed of various variables (attributes). Different instances of a class have the same services, the same behavior and the same variables (attributes).
  However, they can have different variable values, e.g., different "Connection Instances": "Explicit Message", "Poll I/O" or "Bit-Strobe Connection Instance".

- **Attributes**
  The attributes represent data provided by a device via DeviceNet<sup>TM</sup>. The contain the current values of e.g., a configuration or input, such as "Vendor ID", "Device Type" or "Product Name".

- **Service**
  Services can be applied to classes and attributes and perform defined actions, e.g., reading attributes or resetting a class.

- **Behavior**
  The behavior defines how a device responds to external events, e.g., changed process data or as a consequence of internal events, e.g., elapsed timers.

## 11.1.3   Characteristics of DeviceNet™ Devices

DeviceNet™- devices are defined by "Vendor ID" and "Device Type":

- **Vendor ID**          0x28 (40)
- **Device Type**       0x0C (12), Communication Adapter

### 11.1.3.1   Communication Model

#### 11.1.3.1.1 Message Groups

DeviceNet™ messages are divided into different groups to obtain various priorities:

- Message group 1 is used for the exchange of I/O data via I/O messages
- Message group 2 is provided for master/slave applications
- Message group 3 is used to exchange configuration data via explicit communication links
- Message group 4 is reserved for future applications (e.g., "Offline Connection Set")

The so-called connection ID, which determines message priority, is established by the various message groups and the DeviceNet™ station address "MAC ID" that is set on the device.

#### 11.1.3.1.2 Message Types

A distinction is made between two message types for DeviceNet™:

- I/O messages
- Explicit messages

##### 11.1.3.1.2.1  I/O Messages

The messages, primarily input/output data, are sent by a node and can be received and processed by one or more nodes. No protocol data are specified in the data field.

##### 11.1.3.1.2.2  Explicit Messages

Explicit messages are directly transmitted from one node to the other. They consist of a requirement and a reply. As such, services can be directly requested or performed by a different subscriber. The data field contains, among other things, the target address and the service identification. The format of explicit messages is fixed. Explicit messages are used to configure devices or to create a dynamic structure of communication links.

### 11.1.3.2　Data Exchange

Process data are exchanged between scanner and DeviceNet<sup>TM</sup> device by means of the following three mechanisms:

- **Polled I/O Connection**
  Slaves are polled cyclically by the master.

- **Change of Cyclic/State**
  Message are transmitted either cyclically by the master or the slave or in the event of a state change.

- **Bit-Strobe**
  All slaves are polled by the master by means of a command.

## 11.1.4　Process Data and Diagnostic Status

The data are transmitted between master and slave in the form of objects, distinguishing between input and output objects. The structure of the objects is determined by assembly objects, which is used to group attributes of different application objects. Input/output data from different objects can be combined into data blocks and transmitted via a communication link.

### 11.1.4.1　Process Image

The process image is distinguished between input and output process image.

The assembly object provides a statically configured process image in instances 1 through 9.

By setting the "Produced Connection Path" and the "Consumed Connection Path" for individual I/O connections (poll, bit strobe, change of state or change of value), the required process image can be selected.

The structure of the individual instances of the assembly object is described below.

### 11.1.4.1.1 Assembly Instances

Permanently pre-programmed (static) assemblies in the device permit easy and rapid transmission of input and output images from the fieldbus coupler/controller to the master. For this purpose, various assembly instances are provided in the fieldbus coupler/controller:

- **Output 1 ("I/O Assembly Instance 1")**
  The entire output data image is transmitted from the master to the fieldbus controller via the corresponding I/O message connection. The data length corresponds to the quantity of output data in bytes. The analog output data come before the digital output data.

- **Output 2 (“I/O Assembly Instance 2”)**
  The digital output data image is transmitted from the master to the controller via the corresponding I/O message connection. The data length corresponds to the quantity of digital output data and is rounded up to full bytes.

- **Output 3 (“I/O Assembly Instance 3”)**
  The analog output data image is transmitted from the master to the controller via the corresponding I/O message connection. The data length corresponds to the quantity of analog output data in bytes.

- **Input 1 (“I/O Assembly Instance 4”)**
  The entire input data image and one status byte are transmitted to the master via the corresponding I/O message connection. The data length corresponds to the quantity of input data in bytes and one status byte.

- **Input 2 (“I/O Assembly Instance 5”)**
  The digital input data image and one status byte are transmitted to the master via the corresponding I/O message connection. The data length corresponds to the quantity of digital input data and is rounded up to full bytes. In addition, a status byte is attached.

- **Input 3 (“I/O Assembly Instance 6”)**
  The analog input data image and one status byte are transmitted to the master via the corresponding I/O message connection. The data length corresponds to the quantity of analog input data in bytes and one status byte.

- **Input 1 (“I/O Assembly Instance 7”)**
  The entire input data image is transmitted to the master via the corresponding I/O message connection. The data length corresponds to the quantity of input data in bytes.

- **Input 2 (“I/O Assembly Instance 8”)**
  The digital input data image is transmitted to the master via the corresponding I/O message connection. The data length corresponds to the quantity of digital input data and is rounded up to full bytes.

- **Input 3 (“I/O Assembly Instance 9”)**
  The analog input data image is transmitted to the master via the corresponding I/O message connection. The data length corresponds to the quantity of analog input data in bytes.

## 11.1.5    Configuration and Parameterization Using the Object Model

### 11.1.5.1    EDS Files

Features of DeviceNet<sup>TM</sup> devices are documented by the manufacturers in the form of EDS files ("Electronic Data Sheet") and made available to the user.

Structure, content and coding of the EDS files are standardized, allowing configuration via configuration devices from various manufacturers.

| EDS file for fieldbus controller 750-806 | 750-806_1.EDS* |
|---|---|

\*    "_1" indicates that this EDS file is valid for fieldbus controllers with firmware major version 1.

The ESD file is read by the configuration software. Corresponding settings are transmitted.

---

### Information

**Additional Information**
Refer to the software user manuals for the important entries and handling steps for settings in the configuration.

---

### Information

**Additional Information**
The "Open DeviceNet Vendor Association" (ODVA) provides information about the EDS files for all listed manufacturers. http://www.odva.org.
EDS and symbol files to configure the I/O modules are available under order number 750-912 on the Internet at: http://www.wago.com.

---

### 11.1.5.2    Object Model

For network communication, DeviceNet<sup>TM</sup> utilizes an object model in which all functions and data of a device are described.

**General Management Objects**
**("System Support Objects")**

- "Identity Object"
- "Message Router Object"

**Communication Objects for Data Exchange**
**("Communication Objects")**

- "DeviceNet Object"
- "Connection Object"

**Application Objects, to determine device function and/or configuration**

- "Application Object(s)"
- "Assembly Object"
- "Parameter Object"

Communication can be connection based exclusively. For access from the network to the individual objects, connections between the required subscribes must first be established and connection objects set up or activated.

The fieldbus controller supports Bit-Strobe (multicast).

- Consumed Path is changeable (Discrete Output Point Object (0x66) or 0 is valid)

- Produced Path is changeable like a poll connection (if data size is more than 8 bytes, only the first 8 bytes are transmitted)

The data types used in the object model are described below.

Table 57: Object Model Data Types

| Data types | |
|---|---|
| USINT | Unsigned Short Integer (8 bit) |
| UINT | Unsigned Integer (16 bit) |
| USINT | Unsigned Short Integer (8 bit) |
| UDINT | Unsigned Double Integer (32 bit) |
| BOOL | Boolean, True (1) or False (0) |
| STRUCT | Structure of ... |
| ARRAY | Array of ... |

Ranges of input classes may overlap, ranges of output ranges classes also.

**Example:**
class 160 / instance 1 and 2 (USINT) = Class 166 / instance 1 (UINT) or
class 166 / instance 1 and 2 (UINT)   = Class 170 / instance 1 (UDINT).

The DeviceNet[TM] fieldbus controller (750-806) is referred to as "Device" in the tables below.

## 11.1.5.2.1 Object Classes

Defined object classes:

Table 58: Object Classes

| Object | Class | Instance | Description |
|---|---|---|---|
| Identity | 0x01 | 1 | "Device type", "Vendor ID", serial number, etc. |
| Message Router | 0x02 | 1 | Routes explicit messages |
| DeviceNet | 0x03 | 1 | Maintains the physical link to the DeviceNet<sup>TM</sup> network. Allocates/deallocates "Master/Slave Connection Set" |
| Assembly | 0x04 | 9 | Allows data transmission of different objects over a single connection, by combining attributes of different objects |
| | | 10, 11 | Additional assembly as reference to the process image that contains the PFC input/output variables |
| | | 100, 101 | Additional dynamic assembly |
| Connection class | 0x05 | 3 | Allows forwarding of explicit messages |
| Acknowledge handler | 0x2B | 1 | Coordinates receipt of message confirmations. Communicates with the application object. Defines "Acknowledge reception", "Acknowledge timeouts" and "Production retry limit" |
| Coupler configuration object | 0x64 | 1 | Device configuration |
| Discrete input point | 0x65 | 0 ... 255 | Digital input channel objects |
| Discrete output point | 0x66 | 0 ... 255 | Digital output channel objects |
| Analog input point | 0x67 | 0 ... 255 | Analog input channel objects |
| Analog output point | 0x68 | 0 ... 255 | Analog output channel objects |
| Module configuration | 0x80 | 1…65 | Description of connected I/O modules |
| Input Fieldbus Variable USINT | 0xA0 | 1 … 255 | PFC Input Variables |
| Input Fieldbus Variable USINT | 0xA1 | 256 … 510 | PFC Input Variables |
| Input Fieldbus Variable USINT | 0xA2 | 511 … 512 | PFC Input Variables |
| Input Fieldbus Variable UINT | 0xA6 | 1 … 255 | PFC Input Variables |
| Input Fieldbus Variable UINT | 0xA7 | 256 | PFC Input Variables |
| Output Fieldbus Variable UINT | 0xA8 | 1 … 255 | PFC Output Variables |
| Output Fieldbus Variable UINT | 0xA9 | 256 | PFC Output Variables |
| Input Fieldbus Variable UDINT | 0xAA | 1 … 128 | PFC Input Variables |
| Input Fieldbus Variable UDINT | 0xAB | 1 … 128 | PFC Input Variables Starts with 2 bytes offset because the ranges overlap: Class 166 / Instance 2 and 3 (UINT) = Class 171 / Instance 1 (UDINT), etc. |
| Output Fieldbus Variable UDINT | 0xAC | 1 … 128 | PFC Output Variables |
| Output Fieldbus Variable UDINT | 0xAD | 1 … 128 | PFC output variables Starts with 2 bytes offset because the ranges overlap: Class 168 / Instance 2 and 3 (UINT) = Class 173 / Instance 1 (UDINT), etc. |

### 11.1.5.2.1.1 Identity Class (0x01)

#### Instance 0

Table 59: Instance 0

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 0x01 | Revision number of the class definition for the "Identity Object" |

#### Instance 1

Table 60: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Vendor | UINT | 40 (0x28) | Vendor identification |
| 2 | required | get | Device Type | UINT | 12 (0x0C) | Displays the device type |
| 3 | required | get | Product Code | UINT | e.g., 806 (0x326), for the 750-806 | Displays a specific device from a specific manufacturer |
| 4 | required | get | Revision Major/ Minor | Struct: USINT/ USINT | e.g., "020b" (FW02.11( 11)) | Revision of the DeviceNet™ device, which represents the "Identity Object" |
| 5 | required | get | Status | WORD | - | Device status |
| 6 | required | get | Serial_ number | UDINT | - | Serial number |
| 7 | required | get | Product name | SHORT_ STRING (num,char char...) | e.g., "WAGO 750-806 V 1.0)" for the 750-806 | Identification |
| 10 | required | get/set | Heartbeat Interval | USINT | 0 | Interval between 2 heartbeat messages in seconds |

#### Services

Table 61: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0E | Get_Attribute_Single | Returns the content of a specific attribute |
| 0x05 | Reset | Invokes the reset service for the device |

### 11.1.5.2.1.2 Message Router (0x02)

No attributes, no services

### 11.1.5.2.1.3  DeviceNet Object (0x03)

#### Instance 0

Table 62: Instance 0

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 0x02 | Revision number of the class definition for the "Identity Object" |

#### Instance 1

Table 63: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | Optional | get/set | MAC ID | USINT | 0 … 63 | Node address |
| 2 | Optional | get | Baud rate | USINT | 0 … 2 | Baud rate |
| 3 | Optional | get/set | BOI | BOOL | 0/1 | Bus-off Interrupt |
| 4 | Optional | get/set | Bus-Off Counter | USINT | 0 … 255 | Number of times CAN switches to Buss-off status |
| 5 | Optional | get | Allocation Information Allocation Choice Byte Master's ID | Struct of: BYTE, USINT | 0 … 63, 255 | s. MAC ID of Master (from Allocate) |

#### Services

Table 64: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0E | Get_Attribute_Single | Returns the content of a specific attribute |
| 0x10 | Set_Attribute_Single | Modifies the content of a specific attribute |
| 0x4B | Allocate_Master/Slave_Connection | Requests use of the predefined Master/Slave connection |
| 0x4C | Release_Group_2_Identifier_Set | Deletes connections via "Predefined Connection Set" |

## 11.1.5.2.1.4  Assembly Object (0x04)

### Instance 0

Table 65: Instance 0

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 0x02 | Revision number of the class definition for the "Assembly Object" |

### Description of Instances

Table 66: Description of Instances

| Instance ID | Description |
|---|---|
| 1 | Reference to process image:  Analog/digital output data |
| 2 | Reference to process image:  Digital output data |
| 3 | Reference to process image:  Analog output data |
| 4 | Reference to process image:  Analog/digital input data plus status |
| 5 | Reference to process image:  Digital input data plus status |
| 6 | Reference to process image:  Analog input data plus status |
| 7 | Reference to process image:  Analog/digital input data |
| 8 | Reference to process image:  Digital input data |
| 9 | Reference to process image:  Analog input data |
| 10 | Reference to the process image that contains the PFC output variables. |
| 11 | Reference to the process image that contains the PFC input variables. |
| 12 | Reference to process image:  Analog/digital input data plus error code |
| 13 | Reference to process image:  Analog/digital input data plus error code and error argument |
| 14 | Reference to process image:  Analog/digital input data plus error code and error argument, status |

### Instance 1

Table 67: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get/set | Process image | Array of byte | Process image, collection of all output data of the I/O modules |

### Instance 2

Table 68: Instance 2

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get/set | Process image | Array of byte | Process image, collection of all digital output data of the I/O modules |

### Instance 3

Table 69: Instance 3

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get/set | Process image | Array of byte | Process image, collection of all analog output data of the I/O modules |

### Instance 4

Table 70: Instance 4

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get | Process image | Array of byte | Process image, collection of all I/O module input data plus status |

### Instance 5

Table 71: Instance 5

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get | Process image | Array of byte | Process image, collection of all digital input data of the I/O modules plus status |

### Instance 6

Table 72: Instance 6

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get | Process image | Array of byte | Process image, collection of all analog input data of the I/O modules plus status |

## Instance 7

Table 73: Instance 7

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get | Process image | Array of byte | Process image, collection of all I/O module input data |

## Instance 8

Table 74: Instance 8

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get | Process image | Array of byte | Process image, collection of all digital input data of the I/O modules |

## Instance 9

Table 75: Instance 9

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get | Process image | Array of byte | Process image, collection of all Analog I/O module input data |

## Instance 10

Table 76: Instance 10

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | optional | get | PFC output variables | Array of Byte | Process image, collection of all PFC Output Variables |

## Instance 11

Table 77: Instance 11

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | optional | get | PFC input variables | Array of Byte | Process image, collection of all PFC Input Variables |

**(For PFC with software version SW 01.06 or lower):**

**PFC Output (I/O Assembly Instance 10):**

Only the PFC output variables are transmitted via the corresponding input/output communication link. The file length corresponds to the value in Class 100 / Instance 1 / Attribute 101 (BK_FBOUT_ VAR_CNT).

**PFC Input (I/O Assembly Instance 11):**

Only the PFC input variables are transmitted via the corresponding input/output communication link. The file length corresponds to the value in Class 100 / Instance 1 / Attribute 100 (BK_FBIN_ VAR_CNT).

**(For PFC software version SW 01.06 or higher):**

**PFC Output (I/O Assembly Instance 10):**

Only the PFC output variables are transmitted via the corresponding input/output communication link. The file length corresponds to the value in Class 100 / Instance 1 / Attribute 104 (BK_FBOUT_ PLCONLY_VAR_CNT).

The first PFC byte to be transmitted is defined with the value in Class 100 / Instance 1 / Attribute 105 (BK_FBOUT_STARTPLC_VAR_CNT).

**PFC Input (I/O Assembly Instance 11):**

Only the PFC input variables are transmitted via the corresponding input/output communication link. The file length corresponds to the value in Class 100 / Instance 1 / Attribute 102 (BK_FBINP_PLCONLY_ VAR_CNT).

The first PFC byte to be transmitted is defined with the value in Class 100 / Instance 1 / Attribute 103 (BK_FBIN_STARTPLC_VAR_CNT).

### Instance 12

Table 78: Instance 12

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get | Process image + Error code | Array of byte | Process image, collection of all Input data from I/O modules plus error code (Class 100, Instance 1, Attribute 45) |

## Instance 13

Table 79: Instance 13

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get | Process image + Error code + Error Argument | Array of byte | Process image, collection of all Input data from I/O modules plus "Error Code" (Class 100, Instance 1, Attribute 45) plus error code (Class 100, Instance 1, Attribute 46) |

## Instance 14

Table 80: Instance 14

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 3 | Depending on the type of I/O module | get | Process image + Error Code + Error Argument + Status +Terminal diagnostics +Diagnostic value | Array of byte | Process image, collection of all input data from I/O modules plus error code (Class 100, Instance 1, Attribute 45) plus error argument (Class 100, Instance 1, Attribute 46) plus status (Class 100, Instance 1, Attribute 5) plus terminal diagnostics (Class 100, Instance 1, Attribute 6) plus diagnostic value) (Class 100, Instance 1, Attribute 47), the diagnostic value is only valid when indicated in the status that there is a diagnostic message |

## Services

Table 81: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0E | Get_Attribute_Single | Returns the content of a specific attribute |
| 0x10 | Set_Attribute_Single | Changes content of a specific attribute |

## 11.1.5.2.1.4.1   Dynamic Assembly

Two dynamic assembly instances are possible (Instance 100 and 101).

Table 82: Dynamic Assembly

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Number of Members in List | UINT | | Mex. 51 list entries can be combined. |
| 2 | required | get/set | Member List | Array of Struct | | The list of combined data is a field with DeviceNet^TM paths. |
| | | | Member Data Description | UINT | Size in bits | Size of the data |
| | | | Member Path Size | UINT | Size in bytes | Size of the path |
| | | | Member Path * | EPATH | | |
| 3 | required | get/set | Data | Array of Byte | | |

\*   **Description of the member path:**
    0x20 CC 0x24 II 0x30 AA

    CC: Class
    II:   Instance
    AA: Attribute

    The following classes / instances/ attributes are possible:
    •   Class 100, instance 1, attribute 5 (ProcessState)
    •   Class 100, instance 1, attribute 6 (DNS_i_Trmnldia)
    •   Class 101 (Discrete Input Point Object)
    •   Class 102 (Discrete Output Point Object)
    •   Class 103 (Analog Input Point Object)
    •   Class 104 (Analog Output Point Object)
    •   Class 160-173 (PLC variables)

### Class Services

Table 83: Class Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0Eh | Get_Attribute_Single | Used to read an object attribute value. |
| 0x08h | Create | Instantiates an assembly object within a specific class. The response contains the instance number. |

**Instance Services**

Table 84: Instance Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0Eh | Get_Attribute_Single | Used to read an object attribute value |
| 0x10h | Set_Attribute_Single | Modifies an attribute value |
| 0x09h | Delete | Deletes an assembly object |

## 11.1.5.2.1.5  Connection Object (0x05)

**Instance 0**

Table 85: Instance 0

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 0x01 | Revision number of the class definition for the "Connection Object" |

**Description of Instances**

Table 86: Description of Instances

| Instance ID | Description |
|---|---|
| 1 | Describes the explicit messaging connection |
| 2 | Describes the poll I/O connection |
| 3 | Describes the bit-strobe I/O connection |
| 4 | Describes the change of state or cycle I/O connection |
| 5 | Describes the dynamic explicit connection |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | Describes the dynamic I/O connection |
| 11 | |
| 12 | |
| 13 | |
| 14 | |

### Instance 1 (Explicit Messaging)

Table 87: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 1 | Available | get | state | USINT | Status of the object |
| 2 | required | get | instance_ type | USINT | Displays I/O or messaging connection |
| 3 | required | get | transport-Class_trigger | USINT | Defines the connection behavior |
| 4 | required | get | produced_ connection_ id | UINT | In the CAN identifier field when transmitting |
| 5 | required | get | consumed_ connection_ id | UINT | CAN identifier field value that denotes the message that should be transmitted |
| 6 | required | get | initial_ comm_ characteristics | USINT | Defines the message groups across which productions and consumptions associated with this connection occur |
| 7 | required | get | produced_ connection_ size | UINT | Maximum number of bytes that can be transmitted via the connection |
| 8 | required | get | consumed_ connection_ size | UINT | Maximum number of bytes that can be received via this connection |
| 9 | required | get/set | expected_ packet_rate | UINT | Defines the timing of the connection |
| 10, 11 | N/A | get | N/A | N/A | not used |
| 12 | required | get | watchdog_ timeout_ action | USINT | Defines how to handle inactivity / watchdog timeouts |
| 13 | required | get | produced_ connection_ path_length | UINT | Number of bytes in "produced_connection_path" attribute |
| 14 | required | get/set | produced_ connection_ path | Array of USINT | Specifies application objects, whose data are generated in this connection object |
| 15 | required | get | consumed_ connection_ path_ length | UINT | Number of bytes in the attribute "consumed_connection_path" |
| 16 | required | get | consumed_ connection_ path | Array of USINT | Specifies application objects, whose data are received by this connection object |
| 17 | required | get | production_ inhibi t_time | USINT | Defines the minimum time between data transmission |

**Instance 2 (Poll I/O Connection)**

Table 88: Instance 2

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 1 | Available | get | state | USINT | Status of the object |
| 2 | required | get | instance_ type | USINT | Displays I/O or messaging connection |
| 3 | required | get | transport Class_trigger | USINT | Defines the connection behavior |
| 4 | required | get | produced_ connection_ id | UINT | In the CAN identifier field when transmitting |
| 5 | required | get | consumed_ co nnection_ id | UINT | CAN identifier field value that denotes the message that should be transmitted |
| 6 | required | get | initial_ comm _characteri stics | USINT | Defines the message groups across which productions and consumptions associated with this connection occur |
| 7 | required | get | produced_ con nection_ size | UINT | Maximum number of bytes that can be transmitted via the connection |
| 8 | required | get | consumed_ co nnection_ size | UINT | Maximum number of bytes that can be received via this connection |
| 9 | required | get/set | expected_ pa cket_rate | UINT | Defines the timing of the connection |
| 10, 11 | N/A | get | N/A | N/A | not used |
| 12 | required | get | watchdog_ timeo ut_action | USINT | Defines how to handle inactivity / watchdog timeouts |
| 13 | required | get | produced_ con nection_ path_ length | UINT | Number of bytes in "produced_connection_path" attribute |
| 14 | required | get/set | produced_ con nection_ path | Array of USINT | Specifies application objects, whose data are generated in this connection object |
| 15 | required | get | consumed_ co nnection_ path_ length | UINT | Number of bytes in "consumed_connection_path" attribute |
| 16 | required | get/set | consumed_ co nnection_ path | Array of USINT | Specifies application objects, whose data are received by this connection object |
| 17 | required | get | production_ inhibi t_time | USINT | Defines the minimum time between data transmission |

### Instance 3 (Bit-Strobe I/O Connection)

Table 89: Instance 3

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 1 | Available | get | state | USINT | Object state |
| 2 | required | get | instance_ type | USINT | Displays I/O or messaging connection |
| 3 | required | get | transportClass_ trigger | USINT | Defines the connection behavior |
| 4 | required | get | produced_ connection_ id | UINT | In the CAN identifier field when transmitting |
| 5 | required | get | consumed_ co nnection_ id | UINT | CAN identifier field value that denotes the message that should be transmitted |
| 6 | required | get | initial_ comm _characteri stics | USINT | Defines the message groups across which productions and consumptions associated with this connection occur |
| 7 | required | get | produced_ con nection_ size | UINT | Maximum number of bytes that can be transmitted via the connection |
| 8 | required | get | consumed_ co nnection_ size | UINT | Maximum number of bytes that can be received via the connection |
| 9 | required | get/set | expected_ pa cket_rate | UINT | Defines the timing of the connection |
| 10, 11 | N/A | get | N/A | N/A | not used |
| 12 | required | get | watchdog_ timeo ut_action | USINT | Defines how to handle inactivity / watchdog timeouts |
| 13 | required | get | produced_ con nection_ path_ length | UINT | Number of bytes in "produced_connection_path" attribute |
| 14 | required | get | produced_ con nection_ path | Array of USINT | Specifies application objects, whose data are generated in this connection object |
| 15 | required | get | consumed_ co nnection_ path_ length | UINT | Number of bytes in "consumed_connection_path" attribute |
| 16 | required | get | consumed_ co nnection_ path | Array of USINT | Specifies application objects, whose data are received by this connection object |

Table 89: Instance 3

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 17 | required | get | production_ inhibi t_time | USINT | Defines the minimum time between data transmission |

## Instance 4 (Change of State und Cyclic I/O Connection)

Table 90: Instance 4

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 1 | Available | get | state | USINT | Object state |
| 2 | required | get | instance_ type | USINT | Displays I/O or messaging connection |
| 3 | required | get | transportC lass_ trigger | USINT | Defines the connection behavior |
| 4 | required | get | produced_ connection_ id | UINT | In the CAN identifier field when transmitting |
| 5 | required | get | consumed_ co nnection_ id | UINT | CAN identifier field value that denotes the message that should be transmitted |
| 6 | required | get | initial_ comm _characteri stics | USINT | Defines the message groups across which productions and consumptions associated with this connection occur |
| 7 | required | get | produced_ con nection_ size | UINT | Maximum number of bytes that can be transmitted via the connection |
| 8 | required | get | consumed_ connection_ size | UINT | Maximum number of bytes that can be received via the connection |
| 9 | required | get/set | expected_ pa cket_rate | UINT | Defines the timing of the connection |
| 10, 11 | N/A | get | N/A | N/A | not used |
| 12 | required | get | watchdog_ timeo ut_action | USINT | Defines how to handle inactivity / watchdog timeouts |
| 13 | required | get | produced_ con nection_ path_ length | UINT | Number of bytes in "produced_connection_path" attribute |
| 14 | required | get/set | produced_ con nection_ path | Array of USINT | Specifies application objects, whose data are generated in this connection object |

Table 90: Instance 4

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 15 | required | get | consumed_connection_path_length | UINT | Number of bytes in "consumed_connection_path" attribute |
| 16 | required | get | consumed_connection_path | Array of USINT | Specifies application objects, whose data are received by this connection object |
| 17 | required | get/set | production_inhibit_time | USINT | Defines the minimum time between data transmission |

## Services

Table 91: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0E | Get_Attribute_Single | Returns the content of a specific attribute |
| 0x10 | Set_Attribute_Single | Modifies the content of a specific attribute |
| 0x05 | Reset | Invokes the reset service for the device |

The instances are unavailable when the connection is in "non-existent" mode.

## I/O Connection Object State
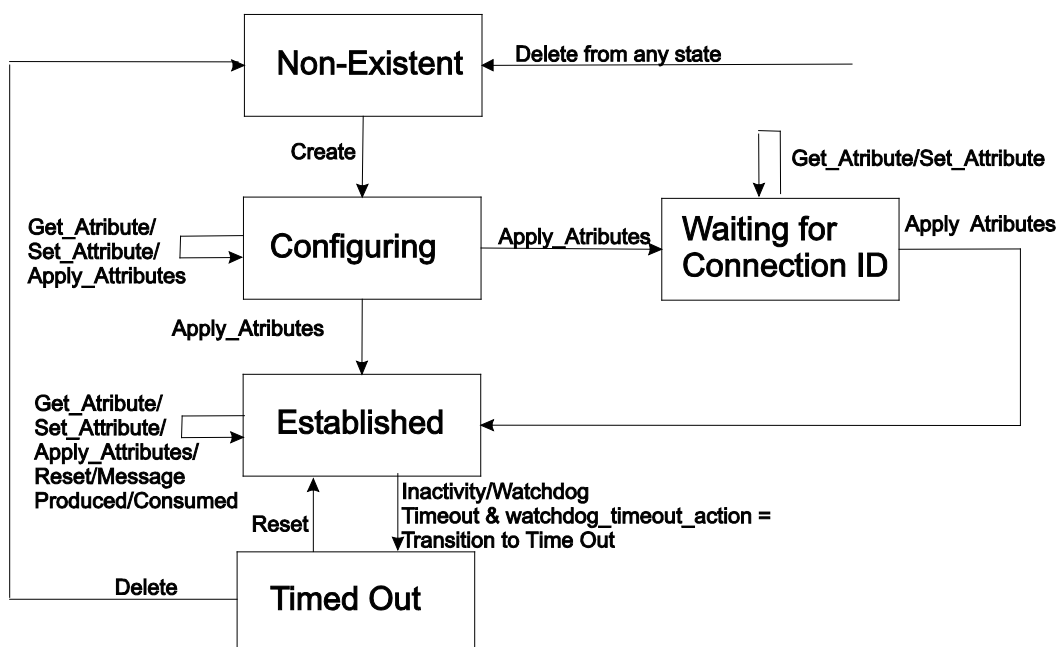


Figure 66: I/O Connection Object State

## 11.1.5.2.1.6  Acknowledge Handler Object (0x2B)

### Instance 0

Table 92: Instance 3

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 0x01 | Revision number of the class definition for the "Acknowledge Handler Object" |
| 2 | required | get | Max instance | UINT | 0x01 | Maximum number of instances of an object generated at the time in this class level of the device |

### Instance 1

Table 93: Instance 4

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 1 | required | get/set | Acknowledge timer | UINT | Time to wait for a confirmation 1 … 65,535 s (0 invalid), default 16 ms |
| 2 | required | get/set | Retry limit | USINT | Number of "Ack Timeouts" elapsed before the application is notified about a "RetryLimit_Reached", event default = 1, range 0 … 255, default 16 ms |
| 3 | required | get | COS Producing Connection Instance | UINT | 0x04, connection instance that contains the path of the I/O application object that is informed by the "Ack Handlers" |

### Services

Table 94: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0E | Get_Attribute_Single | Returns the content of a specific attribute |
| 0x10 | Set_Attribute_Single | Modifies the content of a specific attribute |

### 11.1.5.2.1.7  Coupler Configuration Object (0x64)

#### Instance 0

Table 95: Instance 0

| Attributes ID | Utilization in Device* | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 0x01 | Revision number of the class definition for the "Identity Object" |
| 2 | required | get | Max instance | UINT | 0x01 | Maximum number of instances of an object generated at the time in this class level of the device |

#### Instance 1

Table 96: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 1 | specific | get/set | Bk_ModuleNo | USINT | Module number: 0 – Coupler/controller 1 – First I/O module 2 – Second I/O module |
| 2 | specific | get/set | Bk_TableNo | USINT | Table number: 0 … 256; not all available |
| 3 | specific | get/set | Bk_RegisterNo | USINT | Register number: 0 … 255 for the coupler/controller  (0 … 63 for I/O modules) |
| 4 | specific | get/set | Bk_Data | UINT | Register data, status |
| 5 | specific | get | ProcessState | USINT | Coupler/controller status: 0x01: Internal bus error, 0x02: Coupler/controller configuration error 0x08: Module diagnostic 0x80: Fieldbus error |
| 6 | specific | get | DNS_i_Trmnldia (*) | UINT | Diagnostic, 0x8000, to decode a message, high byte (bit 14 … 8): channel number, low byte (bit 7 … 0) module number |
|  | **(*) Object 100 (0x64) Instance 1 Attribute 6**<br>The "DNS_i_Trmndia" attribute is set depending on the state of the node, i.e., it will execute a diagnostic evaluation. However, this word only supplies valid data if bit 3 (counted up from 0) in "ProcessState" (Class 100, Instance 1, Attribute 5) is set. This bit indicates that a new diagnostic message is available (see "ProcessState" description).<br>The diagnostic evaluation is done by bit 15 in the "DNS_i_Trmndia" attribute. There is a diagnostic error if bit 15 is set. The error is rectified if bit 15 is not set. As long as at least one diagnostic error is present, the MS LED is flashing red. If there are several diagnostic notifications at the same time, the next diagnostic notification is output with every readout of this attribute. If DNS_i_Trmndia = 0, there are currently no new diagnostic notifications. The MS LED only changes back to green after the last diagnostic message has been read (if the diagnostic reason has been resolved). | | | | | |
| 7 | specific | get | CnfLen. AnalogOut | UINT | Number of input/output bits for the analog output data words |

Table 96: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 8 | specific | get | CnfLen. AnalogInp | UINT | Number of input/output bits for the analog input data words |
| 9 | specific | get | CnfLen. DigitalOut | UINT | Number of input/output bits for the digital output bits |
| 10 | specific | get | CnfLen. DigitalInp | UINT | Number of input/output bits for the digital input bits |
| 11 | specific | get/set | BK_FAULT_ REACTION | USINT | Enumeration for error handling 0: Stops local input/output cycles (default) 1: Sets all outputs to 0 2: Does nothing |
| 12 | specific | get/set | BK_SEL_ STORED_POL L_P_PATH | UINT | Power-up value for send path. "Polled I/O" enumeration for the assembly path, class and instance for the object path. (Notice: e.g., do not choose analog inputs if only digital input modules on the node) 4: Analog/digital input data plus status 5: Digital input data plus status 6: Analog input data plus status 7: Analog/digital input data 8: Digital input data 9: Analog input data 12: Analog/digital input data plus BK_LED_ERR_CODE (C 100, I 1, A45) 13: Analog/digital input data plus BK_LED_ERR_CODE (C 100, I 1, A45) plus BK_LED_ERR_ARG (C 100, I 1, A46) 14: Analog/digital input data plus BK_LED_ERR_CODE (C 100, I 1, A45) plus BK_LED_ERR_ARG (C 100, I 1, A46) plus Status (C 100, I 1 A 5) plus DNS_i_Trmnldia (C 100, I 1, A6) plus BK_DIAG_VALUE (C 100, I 1, A47) |

Table 96: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 13 | specific | get/set | BK_SEL_STORED_POLL_C_C_PATH | UINT | Power-up value for the receive path. "Polled I/O" enumeration for the assembly path, class and instance for the I/O module objects. Notice: Only write values to this attribute that are also available, e.g., do not use analog inputs if only digital input/output modules are linked. |
| 14 | specific | get/set | BK_SEL_STORED_COSCYC_C_PATH | UINT | Power-up value for change of state and cyclic connection. Enumeration for the assembly path, class and instance of the I/O module objects. Notice: Only write values to this attribute that are also available, e.g., do not use analog inputs if only digital input/output modules are linked. |
| 15 | specific | get/set | BK_EM_expected_packet_rate | UINT | Defines the default time for the explicit messaging connection |
| 16 | specific | get/set | BK_EM_watchdog_timeout_action | USINT | Defines how inactivity/watchdog explicit messaging connection timeouts are handled |
| 17 | specific | get/set | BK_PIO_expected_packet_rate | UINT | Defines the default time for the poll I/O connection |
| 18 | specific | get/set | BK_PIO_watchdog_timeout_action | USINT | Defines how inactivity/watchdog poll connection timeouts are handled |
| 19 | specific | get/set | BK_BS_expected_packet_rate | UINT | Defines the default time for the bit-strobe I/O connection |
| 20 | specific | get/set | BK_BS_watchdog_timeout_action | USINT | Defines how inactivity/watchdog bit-strobe I/O connection timeouts are handled |
| 21 | specific | get/set | BK_COS_expected_packet_rate | UINT | Defines the default time for the change of state and cyclic I/O connection |
| 22 | specific | get/set | BK_COS_watchdog_timeout_action | USINT | Defines how inactivity/watchdog change of state and cyclic I/O connection timeouts are handled |

Table 96: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 23 | specific | get/set | BK_BOI | USINT | Defines the default value for BIO (Obj. 0x03 Inst.1, Att.3.)   It handles the CAN-Bus-Off situation<br>0: After detection of a bus-off alarm, keep the CAN chip in the Bus-Off (Reset) status status.<br>1: If possible, full reset of the CAN chip and after detection of a bus-off alarm, continue with communication. |
| 24 | specific | get/set | BK_DO_FAULT_REACTION_ON_RELEASE_PIO | USINT | Defines the behavior after allocation of the polled I/O connection<br>0: No activity (default)<br>1: Error behavior processing for fieldbus coupler/controller |
| 25 | specific | get/set | BK_DO_FAULT_REACTION_ON_RELEASE_COS | USINT | Defines the behavior after allocation of the change of state and cyclic<br>I/O connection<br>0: No activity (default)<br>1: Error behavior processing for fieldbus coupler/controller |
| 26 | specific | get/set | BK_DO_FAULT_REACTION_ON_RELEASE_ST | USINT | Defines the behavior after allocation of the strobed connection<br>0: No activity (default)<br>1: Error behavior processing for fieldbus coupler/controller |
| 40 | specific | get/set | BK_static_analog_digital_input_mapping | UINT | Defines how the values for the analog and digital input bits are determined.<br>0000: All bits are digital.<br>0016: One word is analog, the remaining bits are digital.<br>0032: Two words are analog, the remaining bits are digital.<br>...<br>0xFFFF: All bits are handled as "Module Type" (default). |

Table 96: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 41 | specific | get/set | BK_static_analog_digital_input_mapping | UINT | Defines how the values for the analog and digital outputs bits are determined. 0000: All bits are digital. 0016: One word is analog, the remaining bits are digital. 0032: Two words are analog, the remaining bits are digital. ... 0xFFFF: All bits are handled as "Module Type" (default). (If the number of analog bits exceeds the size of the process image, all bits are mapped to analog bits.) |
| 42 | specific | get/set | BK_specific_coupler_behavior | UINT | Defines the fieldbus coupler/controller functionality. 0xFFFF: All possible functions are enabled. (Resetting a bit to 0 disables the related function.) It is only possible to reduce the functionality. Resetting to "1" is ignored. |
| 43 | specific | get/set | BK_revision_setting | UINT | Defines the major and minor revision attributes of the fieldbus coupler/controller. 0xFFFF: The major and minor revision attributes are set by the firmware by default. 0xXX00: The minor revision is set to 0. 0x03XX: The major revision is set to 3. All other values are also valid. |
| 44 | specific | get/set | BK_SAVE_DYN_ASS_INST | UINT | Saves instances created dynamically in the non-volatile memory (after turning on the power supply, all stored instances are created automatically). 0: Saves no dynamic instances 1: Saves dynamic instances |
| 45 | specific | get | BK_LED_ERR_CODE | UINT | Contains the error code displayed using I/O LED blink code. |
| 46 | specific | get | BK_LED_ERR_ARG | UINT | Contains the error argument displayed using I/O LED blink code. |
| 47 | specific | get | BK_DIAG_VALUE | UINT | Contains diagnostic information supplied from the I/O module. This attribute must be read before attribute 6 of this class. This only contains valid data when attribute 6 contains valid data. |

Table 96: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 51 | specific | get/set | BK_FBC_CFG | UINT | Contains configuration settings Bit 0: =1 turns off displaying current diagnostic messages via MS LED. The diagnostic messages, however, are still transmitted via the fieldbus. Any change to this setting is only applied after a power-on cycle. Bit 1..15: reserved |
| 100 (0x64) | specific | get/set | BK_FBINP_ VAR_CNT | UINT | Defines the number of bytes for the PFC fieldbus variables (inputs) to be added to the assembly object (this number is added to the consumption path – Assembly Instances 1 … 3) |
| 101 (0x65) | specific | get/set | BK_FBOUT_ VAR_CNT | UINT | Defines the number of bytes for the PFC fieldbus variables (outputs) to be added to the assembly object (this number is added to the production path – Assembly Instances 4 … 9) |
| **(For PFC with software version SW 01.06 or higher):** | | | | | |
| 102 (0x66) | specific | get/set | BK_FBINP_ PLCONLY_VA R_CNT | UINT | Defines the number of bytes for the PFC fieldbus variables (inputs) assigned to the assembly object "Instance 11" |
| 103 (0x67) | specific | get/set | BK_FBINP_ START_PLC_V AR_CNT | UINT | Defines the offset from the first PFC input variables (the number specified in Attribute 102 is transmitted to PFC input variables from this variable) |
| 104 (0x66) | specific | get/set | BK_FBOUT_ PLCONLY_VA R_CNT | UINT | Defines the number of bytes for the PFC fieldbus variables (outputs) assigned to the assembly object "Instance 10" |
| 105 (0x67) | specific | get/set | BK_FBINP_ START_ PLC_VAR _CNT | UINT | Defines the offset from the first PFC output variables (the number specified in Attribute 104 is transmitted to PFC output variables from this variable) |

**Example:**

This example is described from the perspective of the DeviceNet[TM] fieldbus controller (750-806):

Configuration of the fieldbus controller:       Input process image 12 bytes,
                                        Output process image 10 bytes

➔  BK_FBINP_VAR_CNT = 0; BK_FBOUT_VAR_CNT = 0

Poll Connection:

- 12-byte input process image (produced)
- 10-byte output process image (consumed)

➔ BK_FBINP_VAR_CNT = 4; BK_FBOUT_VAR_CNT = 3

Poll Connection:

- 15 bytes (produced)
  (12-byte input process image + 3-byte PFC output variables)

- 14 bytes (consumed)
  (10-byte output process image + 4-byte PFC input variables)

### Services

Table 97: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0E | Get_Attribute_Single | Returns the content of a specific attribute |
| 0x10 | Set_Attribute_Single | Modifies the content of a specific attribute |

## 11.1.5.2.1.8  Discrete Input Point Object (0x65)

### Instance 0

Table 98: Instance 0

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 0x01 | Revision number of the class definition for the "Identity Object" |
| 2 | optional | get | Max instance | UINT | 0x256 | Maximum number of instances of an object generated at the time in this class level of the device |

### Description of Instances

Table 99: Description of Instances

| Instance ID | Description |
|---|---|
| 1 | Reference to the first digital input point |
| 2 | Reference to the next digital input point |
| … | |
| 255 | Reference to the last possible digital input point |

**Instance 1 … 255**

Table 100: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | Depending on the type of I/O module | get | DIPOBJ_ VALUE | BIT | 0: off 1: on | Digital input bit |

**Services**

Table 101: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0E | Get_Attribute_Single | Returns the content of a specific attribute |

## 11.1.5.2.1.9  Discrete Output Point Object (0x66)

**Instance 0**

Table 102: Instance 0

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 0x01 | Revision number of the class definition for the "Identity Object" |
| 2 | optional | get | Max instance | UINT | 0x256 | Maximum number of instances of an object generated at the time in this class level of the device |

**Description of Instances**

Table 103: Description of Instances

| Instance ID | Description |
|---|---|
| 1 | Reference to the first digital output point |
| 2 | Reference to the next digital output point |
| … | |
| 255 | Reference to the last possible digital output point |

### Instance 1 … 255

Table 104: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | Depending on the type of I/O module | get | DIPOBJ_ VALUE | BIT | 0: off 1: on | Digital output bit |

### Services

Table 105: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0E | Get_Attribute_Single | Returns the content of a specific attribute |
| 0x10 | Set_Attribute_Single | Modifies the content of a specific attribute |

## 11.1.5.2.1.10 Analog Input Point Object (0x67)

### Instance 0

Table 106: Instance 0

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 1 | Revision number of the class definition for the "Identity Object" |
| 2 | optional | get | Max instance | UINT | 256 | Maximum number of instances of an object generated at the time in this class level of the device |

### Description of Instances

Table 107: Description of Instances

| Instance ID | Description |
|---|---|
| 1 | Reference to the first analog input point |
| 2 | Reference to the next analog input point |
| … | |
| 255 | Reference to the last possible analog input point |

### Instance 1 … 255

Table 108: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | Depending on the type of I/O module | get | AIPOBJ_ VALUE | Array of Byte | Current Input values | Input data |
| 2 | Dep. on type of module | get | AIPOBJ_ VALUE | USINT | No. Bytes | Input data lengths |

### Services

Table 109: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0E | Get_Attribute_Single | Returns the content of a specific attribute |
| 0x10 | Set_Attribute_Single | Modifies the content of a specific attribute |

## 11.1.5.2.1.11 Analog Output Point Object (0x68)

### Instance 0

Table 110: Instance 0

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 1 | Revision number of the class definition for the "Identity Object" |
| 2 | optional | get | Max instance | UINT | 256 | Maximum number of instances of an object generated at the time in this class level of the device |

### Description of Instances

Table 111: Description of Instances

| Instance ID | Description |
|---|---|
| 1 | Reference to the first analog output point |
| 2 | Reference to the next analog output point |
| … | |
| 255 | Reference to the last possible analog output point |

### Instance 1 … 255

Table 112: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | Depending on the type of I/O module | get | AOPOBJ_ VALUE | Array of Byte | Current Output values | Output data |
| 2 | Depending on the type of I/O module | get | AOPOBJ_ VALUE | USINT | No. Bytes | Output data lengths |

### Services

Table 113: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0E | Get_Attribute_Single | Returns the content of a specific attribute |
| 0x10 | Set_Attribute_Single | Modifies the content of a specific attribute |

## 11.1.5.2.1.12 Module Configuration Object (0x80)

### Instance 0

Table 114: Instance 0

| Attributes ID | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|
| 1 | get | Revision | UINT | 1 | Revision number of the class definition for the "Identity Object" |
| 2 | get | Max instance | UINT | - | Maximum number of instances of an object generated at the time in this class level of the device |

### Description of Instances

Table 115: Description of Instances

| Instance ID | Description |
|---|---|
| 1 | Reference to the first I/O module |
| … | |
| 65 | Reference to the last possible I/O module |

**Instance 1 … 65**

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Description |
|---|---|---|---|---|---|
| 1 | Depending on the type of I/O module | get | AOPOBJ_ VALUE | Array of Byte | Description of the connected I/O module (Instance 1 = Fieldbus coupler/controller) Bit 0: I/O module has inputs Bit 1: I/O module has outputs Bit 8…14: Internal data width (bit) Bit 15: 0/1 Analog input/output modules / digital input/output modules For analog input/output modules, bits 0…14 designate the I/O module type, e.g., 467 for I/O module 750-467 |

## 11.1.5.2.1.13 Input Fieldbus Variable (0xA0)

**Instance 0**

Table 116: Instance 0

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | required | get | Revision | UINT | 0x01 | Revision number of the object |
| 2 | optional | get | Max instance | UINT | 255 | Max. instance number of an object that is currently created in this class of device |

**Description of Instances**

Table 117: Description of Instances

| Instance ID | Description |
|---|---|
| 1 | Reference to the first PFC input variable byte |
| 2 | Reference to the next PFC input variable byte |
| … | |
| 255 | Reference to the 255th PFC input variable byte |

**Instance 1 … 255**

Table 118: Instance 1

| Attributes ID | Utilization in Device | Access Rights | Name | Data Type | Value | Description |
|---|---|---|---|---|---|---|
| 1 | optional | get/set | FB_IN_ VAR | USINT | Input data | Current input data |

**Services**

Table 119: Service

| Service Code | Service Name | Description |
|---|---|---|
| 0x0Eh | Get_Attribute_Single | Used to read an object attribute value |
| 0x10h | Set_Attribute_Single | Used to write an object attribute value |

### 11.1.5.2.1.14 Input Fieldbus Variable (0xA1)

Class 161

PFC input variable bytes 256 … 510

Max. instance: 255

USINT

### 11.1.5.2.1.15 Input Fieldbus Variable (0xA2)

Class 162

PFC input variable bytes 511 … 512

Max. instance: 2

USINT

### 11.1.5.2.1.16 Input Fieldbus Variable (0xA6)

Class 166

PFC input variable byte 1 … 255

Max. instance: 255

UINT

### 11.1.5.2.1.17 Input Fieldbus Variable (0xA7)

Class 167

PFC input variable byte 256

Max. instance: 1

UINT

### 11.1.5.2.1.18 Output Fieldbus Variable (0xA8)

Class 168

PFC output variable byte 1 … 255

Max. instance: 255

UINT

### 11.1.5.2.1.19 Output Fieldbus Variable (0xA9)

Class 169

PFC output variable byte 256

Max. instance: 1

UINT

### 11.1.5.2.1.20 Input Fieldbus Variable (0xAA)

Class 170

PFC input variable byte 1 … 128

Max. instance: 128

UDINT

### 11.1.5.2.1.21 Input Fieldbus Variable (0xAB)

Class 171

PFC input variable byte 1 … 128

Max. instance: 128

UDINT

Starts with 2 bytes offset because the ranges overlap: Class 166 / Instance 2 and 3 (UINT) = Class 171 / Instance 1 (UDINT), etc.

### 11.1.5.2.1.22 Output Fieldbus Variable (0xAC)

Class 172

PFC output variable byte 1 … 128

Max. instance: 128

UDINT

### 11.1.5.2.1.23 Output Fieldbus Variable (0xAD)

Class 173

PFC output variable byte 1 … 128

Max. instance: 128

UDINT

Starts with 2 bytes offset because the ranges overlap:

Class 168 / Instance 2 and 3 (UINT) = Class 173 / Instance 1 (UDINT), etc.

# 12    I/O Modules

## 12.1    Overview

For modular applications with the WAGO-I/O-SYSTEM 750/753, different types of I/O modules are available

- Digital Input Modules

- Digital Output Modules

- Analog Input Modules

- Analog Output Modules

- Specialty Modules

- System Modules

For detailed information on the I/O modules and the module variations, refer to the manuals for the I/O modules.

You will find these manuals on the WAGO web pages under www.wago.com.

---

**Information**

i

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

---

# 12.2    Process Data Architecture for DeviceNet[TM]

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

With the DeviceNet[TM] fieldbus controller, the process image uses a byte structure (no word alignment). The internal mapping method for data greater than one byte conforms to Intel formats.

The fieldbus-specific representation of compatible I/O modules in the process image of the DeviceNet[TM] fieldbus controller is described based on examples.

> ## Note
> **Take into account the process data of all connected I/O modules!**
> 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

The digital input modules provide a process value of one bit per channel, which indicates the status of the respective channel. These bits are mapped in the input process image.

Each input channel occupies one instance in the "Discrete Input Point Object" (Class 0x65). I/O modules with data bits assigned by channel also occupy more instances of this class accordingly.

**Example: 2-channel digital input modules with diagnostics (4 instances)**

Table 120: 2-Channel Digital Input Modules with Diagnostics (4 Instances)

| 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 |

These input modules occupy 4 instances in the "Discrete Input Point Object" (0x65) class.
Some digital input modules provide additional data, e.g., acknowledgements mapped in the output process image. The output data of these I/O modules occupies a corresponding number of instances in class (0x66).

170    I/O Modules
           WAGO-I/O-SYSTEM 750
           750-806 Fieldbus Controller DeviceNet™

**Example: 8-channel digital input modules with diagnostics (8 instances)**

Table 121: 8-Channel Digital Input Modules (8 Instances)

| 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 4 | Data bit DI 2 Channel 2 | Data bit DI 1 Channel 1 |

These input modules occupy 4 instances in the "Discrete Input Point Object" (0x65) class.

## 12.2.2   Digital Output Modules

The digital output modules provide one bit as the process value per channel that indicates the status of the respective channel. These bits are mapped in the output process image.

Each output channel occupies one instance in the "Discrete Output Point Object" (Class 0x66). I/O modules with data bits assigned by channel also occupy more instances of this class accordingly.

Some digital output modules provide additional data, e.g., status bits mapped in the input process image. The input data of these I/O modules occupies a corresponding number of instances in class (0x65).

For 4-channel or 8-channel digital output modules with diagnostics, the diagnostic bits are occupied accordingly in the input process image.

**Example: 8-channel digital output modules with diagnostics (8 instances)**

Table 122: 8-Channel Digital Output Modules with Diagnostics (8 Instances)

| Input Process Image | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| Diagnostic bit S 7 Channel 8 | Diagnostic bit S 6 Channel 7 | Diagnostic bit S 5 Channel 6 | Diagnostic bit S 4 Channel 5 | Diagnostic bit S 3 Channel 4 | Diagnostic bit S 2 Channel 4 | Diagnostic bit S 1 Channel 2 | Diagnostic bit S 0 Channel 1 |

Diagnostic bit S = '0'       no error
Diagnostic bit S = '1' wire break, short circuit or overload

These input modules occupy 8 instances in the "Discrete Input Point Object" (0x65) class.

Table 123: 8-Channel Digital Output Modules with Diagnostics (8 Instances)

| 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 4 | controls DO 2 Channel 2 | controls DO 1 Channel 1 |

These input modules occupy 8 instances in the "Discrete Input Point Object" (0x66) class.

## 12.2.3    **Analog Input Modules**

The analog input modules provide 16-bit measured data and 8 control/status bits per channel. DeviceNet^{TM} does not use the 8 control/status bits, however, i.e., it does not access or evaluate them.
Therefore, only the 16-bit measured values per channel are in Intel format and mapped by byte in the input process image for the fieldbus for the DeviceNet^{TM} fieldbus controller.

If digital input modules are in the node, the analog input data is mapped first into the input process image, followed by the digital input data.

Each input channel occupies one instance in the "Analog Input Point Object" (Class 0x67).

### Example: 8-channel analog input modules (8 instances)

Table 124: 8-Channel Analog Input Modules (8 Instances)

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | D0 | Measured value channel 1 |
| | D1 | |
| n+1 | D2 | Measured value channel 2 |
| | D3 | |
| n+2 | D4 | Measured value channel 3 |
| | D5 | |
| n+3 | D6 | Measured value channel 4 |
| | D7 | |
| n+4 | D8 | Measured value channel 5 |
| | D9 | |
| n+5 | D10 | Measured value channel 6 |
| | D11 | |
| n+6 | D12 | Measured value channel 7 |
| | D13 | |
| n+7 | D14 | Measured value channel 8 |
| | D15 | |

These input modules represent 8 x 2 bytes and occupy 8 instances in class (0x67).

Particularly interesting are I/O modules that send multiple measured values per channel, e.g., the 1-channel analog input module 750-491 that provides two measured values per channel.

## 12.2.4   Analog Output Modules

The analog output modules provide 16-bit output values and 8 control/status bits per channel. DeviceNet<sup>TM</sup> does not use the 8 control/status bits, however, i.e., it does not access or evaluate them.

Therefore, only the 16-bit output values for each channel are in Intel format and are mapped by byte in the output process image for the fieldbus with the DeviceNet<sup>TM</sup> fieldbus controller.

If digital output modules are in the node, the analog output data is mapped first into the output process image, followed by the digital output data.

Each output channel occupies one instance in the "Analog Output Point Object" (Class 0x68).

**Example: 2-channel output modules**

Table 125: 2-channel analog output modules

| Output process image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | D0 | Output value channel 1 |
| | D1 | |
| n+1 | D2 | Output value channel 2 |
| | D3 | |

These output modules represent 2 x 2 bytes and occupy 2 instances in class (0x68).

## 12.2.5   Specialty Modules

In addition to the data bytes, the control/status byte is also displayed for select I/O modules. This byte is used for the bi-directional data exchange of the I/O module with the higher-level control system.

The control byte (Control "C") is transmitted from the controller to the I/O module. The status byte (Status "S") is transmitted from the I/O module to the controller. As a result, it is possible to set the counter with the control byte or indicate a range overflow/underflow with the status byte.

---

**Information**

**Additional Information**
The special configuration of the respective control/status byte can be found in the corresponding description of the I/O module. A manual with the detailed description of each I/O module can be found on the Internet at:
http://www.wago.com.

---

The specialty modules represent analog I/O modules. Therefore, their process input values also occupy one instance per channel in the "Analog Input Point Object" (Class 0x67) and their process output values one instance per channel in the "Analog Output Point Object" (Class 0x68).

**Up/down counters**
750-404, (and all versions except /000-005),
753-404, (and versions /000-003)

In the input and output process image, counter modules occupy 5 bytes of user data: 4 data bytes and 1 additional control/status byte. The I/O module provides 32-bit counter values. 6 bytes are occupied in the process image.

Table 126: Input Process Image Counter Modules

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S | Status byte |
| | - | not used |
| | D0 | Counter value |
| | D1 | |
| | D2 | |
| | D3 | |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x67).

Table 127: Output Process Image Counter Modules

| Output process image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C | Control byte |
| | - | not used |
| | D0 | Counter setting value |
| | D1 | |
| | D2 | |
| | D3 | |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class
(0x68).

## Up/down counters
750-404/000-005

In the input and output process image, counter modules occupy 5 bytes of user
data: 4 data bytes and 1 additional control/status byte. The I/O module provides
16-bit counter values for every counter. 6 bytes are occupied in the process image.

Table 128: Input Process Image, Counter Modules

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S | Status byte |
| | - | not used |
| | D0 | Counter value of counter 1 |
| | D1 | |
| | D2 | Counter value of counter 2 |
| | D3 | |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class
(0x67).

Table 129: Output Process Image, Counter Modules

| Output process image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C | Control byte |
| | - | not used |
| | D0 | Counter setting value counter 1 |
| | D1 | |
| | D2 | Counter setting value counter 2 |
| | D3 | |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class
(0x68).

### 2-channel pulse width output module, 24 VDC
750-511 (and all versions)

In the input and output process image, pulse width output modules occupy 6 bytes of user data: 4 data bytes and two additional control/status bytes. 6 bytes are occupied in the process image.

Table 130: Input/Output Process Image, Pulse Width Output Modules

| Input and Output Process Image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| n | C0/S0 | Control/status byte of channel 1 |
| | D0 | Data value of channel 1 |
| | D1 | |
| n+1 | C1/S1 | Control/status byte of channel 2 |
| | D2 | Counter setting value of channel 2 |
| | D3 | |

These specialty modules represent 2 x 3 bytes and occupy 2 instances in class (0x67) and 2 instances in class (0x68).

### 2-channel up/down counters
750-638, 753-638

In the input and output process image, counter modules occupy 6 bytes of user data: 4 data bytes and two additional control/status bytes. The I/O module provides 16-bit counter values for every counter. 6 bytes are occupied in the process image.

Table 131: Input Process Image Counter Modules

| Input Process Image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| n | S0 | Status byte of counter 1 |
| | D0 | Counter value of counter 1 |
| | D1 | |
| n+1 | S1 | Status byte of counter 2 |
| | D2 | Counter value of counter 2 |
| | D3 | |

These specialty modules represent 2 x 3 bytes and occupy 2 instances in class (0x67).

Table 132: Output Process Image Counter Modules

| Output process image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C0 | Control byte of counter 1 |
| | D0 | Counter setting value counter 1 |
| | D1 | |
| n+1 | C1 | Control byte of counter 2 |
| | D2 | Counter setting value counter 2 |
| | D3 | |

These specialty modules represent 2 x 3 bytes and occupy 2 instances in class (0x68).

### SSI transmitter interface
750-630 (and all versions /000-001, -002, -006, -008, -009, -011, -012, -013)

In the input and output process image, SSI transmitter interface modules occupy 4 data bytes. A total of 4 bytes are occupied in the process image.

Table 133: Input Process Image, SSI Transmitter Interface Modules

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | D0 | Data bytes |
| | D1 | |
| n+1 | D2 | |
| | D3 | |

These specialty modules represent 2 x 2 bytes and occupy 2 instances in class (0x67).

### SSI transmitter interface
750-630/000-004, -005, -007

In the input process image, SSI transmitter interface modules with status occupy 5 bytes of user data: 4 data bytes and one additional status byte. A total of 6 bytes are occupied in the process image.

Table 134: Input Process Image, SSI Transmitter Interface Modules

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S | Status byte |
| | - | not used |
| | D0 | Data bytes |
| | D1 | |
| | D2 | |
| | D3 | |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x67).

### Incremental encoder interface
750-631

The I/O module 750-631 occupies 5 bytes in the input process image and 3 bytes in the output process image. 6 bytes are occupied in the process image.

Table 135: Input Process Image, Distance and Angle Measurement

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S | Status byte |
| | D0 | Counter word |
| | D1 | |
| | - | not used |
| | D2 | Latch word |
| | D3 | |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x67).

Table 136: Output Process Image, Distance and Angle Measurement

| Output Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C | Control byte |
| | D0 | Counter setting word |
| | D1 | |
| | - | not used |
| | - | |
| | - | |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x68).

### Proportional valve module
750-632

The proportional valve module appears in 1-channel operation (1 valve) with 6 bytes, and in 2-channel operation (2 valves) with 12 bytes.

### 1-channel mode

Table 137: Input Process Image, Proportional Valve Module

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S0 | Status byte |
| | MBX_ST | Mailbox status byte |
| | MBX_DATA | Mailbox data |
| | V1_STATUS | Valve 1 control |
| | V1_ACTUAL_L | Valve 1, actual value, low byte |
| | V1_ACTUAL_H | Valve 1, actual value, high byte |

Table 138: Output Process Image, Proportional Valve Module

| Output process image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C0 | Control byte |
| | MBX_CTRL | Mailbox control byte |
| | MBX_DATA | Mailbox data |
| | V1_CONTROL | Valve 1 control |
| | V1_SETPOINTVALUE_L | Valve 1, target value, low byte |
| | V1_SETPOINTVALUE_H | Valve 1, target value, high byte |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

## 2-channel mode

Table 139: Input Process Image, Proportional Valve Module

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S0 | Status byte |
| | MBX_ST | Mailbox status byte |
| | MBX_DATA1 | Mailbox data |
| | MBX_DATA2 | |
| | MBX_DATA3 | |
| | MBX_DATA4 | |
| | V1_STATUS | Valve 1 control |
| | V2_STATUS | Valve 2 control |
| | V1_ACTUAL_L | Valve 1, actual value, low byte |
| | V1_ACTUAL_H | Valve 1, actual value, low byte |
| | V2_ACTUAL_L | Valve 2, actual value, low byte |
| | V2_ACTUAL_H | Valve 2, actual value, low byte |

Table 140: Output Process Image, Proportional Valve Module

| Output process image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C0 | Control byte |
| | MBX_CTRL | Mailbox control byte |
| | MBX_DATA1 | Mailbox data |
| | MBX_DATA2 | |
| | MBX_DATA3 | |
| | MBX_DATA4 | |
| | V1_CONTROL | Valve 1 control |
| | V2_CONTROL | Valve 2 control |
| | V1_SETPOINTVALUE_L | Valve 1, target value, low byte |
| | V1_SETPOINTVALUE_H | Valve 1, target value, high byte |
| | V2_SETPOINTVALUE_L | Valve 2, target value, low byte |
| | V2_SETPOINTVALUE_H | Valve 2, target value, high byte |

These specialty modules represent 1 x 12 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

In Class 0x4, Instance 0x7 of the assembly object, the analog and digital input data occupy 6 or 12 bytes. The output data occupy 6 or 12 bytes in Class 0x4, Instance ID 0x1.

### Incremental encoder interface
750-634

The incremental encoder interface module occupies 5 bytes in the input process image, or 6 bytes in cycle duration measurement operating mode, and 3 bytes in the output process image. 6 bytes are occupied in the process image.

Table 141: Input Process Image, Incremental Encoder Interface Module

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S | Status byte |
| | D0 | Counter word |
| | D1 | |
| | D2*) | (Cycle duration) |
| | D3 | Latch word |
| | D4 | |

 *) If the control byte sets the operating mode to cycle duration measurement, D2 together with D3/D4 provides a 24-bit value for the cycle duration.

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x67).

Table 142: Output Process Image, Incremental Encoder Interface Module

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C | Control byte |
| | D0 | Counter setting word |
| | D1 | |
| | - | not used |
| | - | |
| | - | |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x68).

### Digital impulse interface
750-635, 753-635

In the input and output process image, the digital impulse interface module occupies a total of 4 bytes of user data: 3 data bytes and 1 additional control/status byte. 4 bytes are occupied in the process image.

Table 143: Input/Output Process Image, Digital Impulse Interface

| Input and Output Process Image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| n | C0/S0 | Control/status byte |
| | D0 | Data bytes |
| | D1 | |
| | D2 | |

These specialty modules represent 1 x 4 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

## DC drive controller
750-636

The I/O module occupies 6 bytes of input and output data in the process image. The position data to be sent and received is stored in 4 output bytes and 4 input bytes. 2 control/status bytes are used to control the module and drive. In addition to the position data in the input process image, extended status information can also be shown.

Table 144: DC Drive Controller Input Process Image

| Input Process Image | | | | |
|---|---|---|---|---|
| Instance | Byte designation | | Remark | |
| n | S0 | | Status byte S0 | |
| | S1 | | Status byte S1 | |
| | D0 | S2 | Actual position (LSB) | Ext. status byte S2 |
| | D1 | S3 | Actual position | Ext. status byte S3 |
| | D2 | S4 | Actual position | Ext. status byte S4 |
| | D3 | S5 | Actual position (MSB) | Ext. status byte S5 |

Table 145: DC Drive Controller Output Process Image

| Output process image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| n | C0 | Control byte C0 |
| | C1 | Control byte C1 |
| | D0 | Setpoint position (LSB) |
| | D1 | Setpoint position |
| | D2 | Setpoint position |
| | D3 | Setpoint position (MSB) |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

### Incremental encoder interface
750-637

The incremental encoder interface module appears with 6 bytes of reference data in the input and output area of the process image, 4 data bytes and two additional control/status bytes. 6 bytes are occupied in the process image.

Table 146: Input/Output Process Image, Incremental Encoder Interface Module

| Input and Output Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C0/S0 | Control/status byte 1 |
| | D0 | Data values |
| | D1 | |
| n+1 | C1/S1 | Control/status byte 2 |
| | D2 | Data values |
| | D3 | |

These specialty modules represent 2 x 3 bytes and occupy 2 instances in class (0x67) and 2 instances in class (0x68).

### RTC module
750-640

In both the input and output process image, the RTC module occupies 6 bytes of user data: 4 data bytes and 1 additional control/status byte, as well as 1 command byte (ID) each. Six bytes are occupied in the process image with word alignment.

Table 147: Input/Output Process Image, RTC Module

| Input and Output Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C/S | Control/status byte |
| | ID | Command byte |
| | D0 | Data bytes |
| | D1 | |
| | D2 | |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

**DALI/DSI master module**
750-641

In the input and output process image, the DALI/DSI master module occupies 6 data bytes, 5 data bytes and 1 additional control/status byte. 6 bytes are occupied in the process image.

Table 148: Input Process Image, DALI/DSI Master Module

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S | Status byte |
| | D0 | DALI response |
| | D1 | DALI response |
| | D2 | Message 3 |
| | D3 | Message 2 |
| | D4 | Message 1 |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x67).

Table 149: Output Process Image DALI/DSI Master Module

| Output process image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S | Control byte |
| | D0 | DALI command, DSI dimming value |
| | D1 | DALI address |
| | D2 | Parameter 2 |
| | D3 | Parameter 1 |
| | D4 | Command extension |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x68).

**Radio receiver I/O module**
750-642

In the input and output process image, the radio receiver I/O module occupies 4 bytes of user data: 3 data bytes and 1 additional control/status byte. However, the 3 bytes of output data are not used. 4 bytes are occupied in the process image.

Table 150: Input Process Image, Radio Receiver I/O Module

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S | Status byte |
| | D0 | |
| n+1 | D1 | Data bytes |
| | D2 | |

Table 151: Output Process Image, Radio Receiver I/O Module

| Output process image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| n | C | Control byte |
| | - | not used |
| n+1 | - | |
| | - | |

These specialty modules represent 2 x 2 bytes and occupy 2 instances in class (0x67) and 2 instances in class (0x68).

## MP Bus Master Module
750-643

In the input and output process image, the MP bus master module occupies 8 bytes of user data: 6 data bytes and 2 additional control/status bytes. 8 bytes are occupied in the process image.

Table 152: Input/Output Process Image, MP Bus Master Module

| Input/Output Process Image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| n | C0/S0 | Control/status byte |
| | C1/S1 | Additional control/status byte |
| | D0 | Data bytes |
| | D1 | |
| | D2 | |
| | D3 | |
| | D4 | |
| | D5 | |

These specialty modules represent 1 x 8 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

## *Bluetooth*® RF transceiver
750-644

The length of the process image of the *Bluetooth*® I/O module can be adjusted to a fixed size of 12, 24 or 48 Byte.

It consists of one control byte (input) or one status byte (output), one empty byte, one 6-, 12- or 18-byte overlayable mailbox (mode 2) and the *Bluetooth*® process data with a size of 4 to 46 bytes.

The *Bluetooth*® I/O module uses between 12 to 48 bytes in the process image. The size 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*® RF Transceiver.

750-644.

Table 153: Output Process Image, *Bluetooth*® RF Transceiver

| Input/Output Process Image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| n | C0/S0 | Control/status byte |
| | - | not used |
| | D0 | Mailbox (0, 6, 12 or 18 bytes) and process data (4 ... 46 bytes) |
| | D1 | |
| | D2 | |
| | … | |
| | D46 | |

The I/O module 750-644 represents a specialty module. Its process data (12, 24 or 48 bytes) occupy one instances in classes 0x67 and 0x68.

### 2-channel vibration velocity/bearing condition monitoring VIB I/O
750-645

In both input and the output process image, the vibration velocity/bearing condition monitoring VIB I/O module occupies 12 bytes of user data: 8 data bytes and 4 additional control/status bytes. 12 bytes are occupied in the process image.

Table 154: Vibration Velocity/Bearing Condition Monitoring VIB I/O

| Input/Output Process Image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| n | C0/S0 | Control/status byte (log. channel 1, sensor input 1) |
| | D0 | Data bytes (log. channel 1, sensor input 1) |
| | D1 | |
| n+1 | C1/S1 | Control/status byte (log. channel 2, sensor input 2) |
| | D2 | Data bytes (log. channel 2, sensor input 2) |
| | D3 | |
| n+2 | C2/S2 | Control/status byte (log. channel 3, sensor input 1) |
| | D4 | Data bytes (log. channel 3, sensor input 1) |
| | D5 | |
| n+3 | C3/S3 | Control/status byte (log. channel 4, sensor input 2) |
| | D6 | Data bytes (log. channel 4, sensor input 2) |
| | D7 | |

These specialty modules represent 4 x 3 bytes and occupy 4 instances in class (0x67) and 4 instances in class (0x68).

## DALI Multi-Master module
753-647

The DALI Multi-Master module occupies a total of 24 bytes in the input and output process image.

The DALI Multi-Master module can be operated in "Easy" mode (default) and "Full" mode. "Easy" mode is used to transmit simply binary signals for lighting control. Configuration or programming via DALI master module is unnecessary in "Easy" mode.

Changes to individual bits of the process image are converted directly into DALI commands for a pre-configured DALI network. 22 bytes of the 24-byte process image can be used directly for switching of ECGs, groups or scenes in the Easy mode. Switching commands are transmitted via DALI and group addresses, where each DALI and each group address is represented by a 2-bit pair.

The structure of the process data is described in detail in the following tables.

Table 155: Overview of Input Process Image in the "Easy" Mode

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | S | Status, active broadcast:<br>Bit 0: 1-/2-button mode<br>Bit 2: Broadcast status ON/OFF<br>Bit 1,3-7:   - |
| | - | reserved |
| | DA0…DA3 | Bit pair for DALI address DA0:<br>Bit 1:    Bit set = ON<br>            Bit not set = OFF<br>Bit 2:    Bit set = Error<br>            Bit not set = No error<br>Bit pairs DA1 to DA63 similar to DA0. |
| | DA4…DA7 | |
| | DA8…DA11 | |
| | DA12…DA15 | |
| | DA16…DA19 | |
| | DA20…DA23 | |
| | DA24…DA27 | |
| | DA28…DA31 | |
| | DA32…DA35 | |
| | DA36…DA39 | |
| | DA40…DA43 | |
| | DA44…DA47 | |
| | DA48…DA51 | |
| | DA52…DA55 | |
| | DA56…DA59 | |
| | DA60…DA63 | |
| | GA0…GA3 | Bit pair for DALI group address GA0:<br>Bit 1:  Bit set = ON<br>          Bit not set = OFF<br>Bit 2:  Bit set = Error<br>          Bit not set = No error |
| | GA4…GA7 | |
| | GA8…GA11 | |
| | GA12…GA15 | |

Table 155: Overview of Input Process Image in the "Easy" Mode

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| | - | Not used |
| | - | |

DA = DALI address
GA = Group address

Table 156: Overview of the Output Process Image in the "Easy" mode

| Input Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C | Broadcast ON/OFF and activate: Bit 0: Broadcast ON Bit 1: Broadcast OFF Bit 2: Broadcast ON/OFF/dimming Bit 3: Broadcast short ON/OFF Bit 4…7: reserved |
| | - | reserved |
| | DA0…DA3 | Bit pair for DALI address DA0: Bit 1:    short: DA switch ON    long: dimming, brighter Bit 2:    short: DA switch OFF    long: dimming, darker Bit pairs DA1 to DA63 similar to DA0. |
| | DA4…DA7 | |
| | DA8…DA11 | |
| | DA12…DA15 | |
| | DA16…DA19 | |
| | DA20…DA23 | |
| | DA24…DA27 | |
| | DA28…DA31 | |
| | DA32…DA35 | |
| | DA36…DA39 | |
| | DA40…DA43 | |
| | DA44…DA47 | |
| | DA48…DA51 | |
| | DA52…DA55 | |
| | DA56…DA59 | |
| | DA60…DA63 | |
| | GA0…GA3 | Bit pair for DALI group address GA0: Bit 1:    short: GA switch ON    long: dimming, brighter Bit 2:    short: GA switch OFF    long: dimming, darker Bit pairs GA1 to GA15 similar to GA0. |
| | GA4…GA7 | |
| | GA8…GA11 | |
| | GA12…GA15 | |
| | Bit 0…7 | Switch to scene 0…15 |
| | Bit 8…15 | |

DA = DALI address
GA = Group address

These specialty modules represent 1 x 24 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

### LON® FTT module
753-648

The process image of the LON® FTT module consists of a control/status byte and 23 bytes of bidirectional communication data that is processed by the WAGO-I/O-*PRO* function block "LON_01.lib". This function block is required for the function of the LON® FTT module and makes a user interface available on the control side.

### Serial interfaces with alternative data format
750-650, (and the versions /000-002, -004, -006, -009, -010, -011, -012, -013)
750-651, (and the versions /000-002, -003)
750-653, (and the versions /000-002, -007)

→ **Note**

**The process image of the version with adjustable operating mode corresponds to the process image of the respective standard I/O module!**
The operating mode of the configurable /003-000 serial interface versions can be set with WAGO-I/O-Check. Based on the operating mode, the process image of these I/O modules is then the same as that of the respective version.

In the input and output process image, serial interface modules using an alternative data format occupy 4 bytes of user data: 3 data bytes and 1 additional control/status byte. 4 bytes are occupied in the process image.

Table 155: Input/Output Process Image of the Serial Interfaces

| Input/Output Process Image | | |
|---|---|---|
| **Instance** | **Byte designation** | **Remark** |
| n | C/S | Control/status byte |
| | D0 | Data bytes |
| n+1 | D1 | |
| | D2 | |

These specialty modules represent 2 x 2 bytes and occupy 2 instances in class (0x67) and 2 instances in class (0x68).

### Serial interfaces with standard data format
750-650/000-001, -014, -015, -016
750-651/000-001,
750-653/000-001, -006

In the input and output process image, serial interface modules using the standard data format occupy 6 bytes of user data: 5 data bytes and 1 additional control/status byte. 6 bytes are occupied in the process image.

Table 156: Input/Output Process Image of the Serial Interfaces with Standard Data Format

| Instance | Byte designation | Remark |
|---|---|---|
| **Input/Output Process Image** | | |
| | C/S | Control/status byte |
| | D0 | |
| | D1 | |
| n | D2 | Data bytes |
| | D3 | |
| | D4 | |

These specialty modules represent 1 x 6 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

### Serial interface RS-232 / RS-485
750-652

**Serial transmission mode:** The data to be sent and received is stored in up to 46 input/output bytes of the process image in serial transmission mode. The data flow is controlled with the control/status byte. The input bytes form the memory area for up to 47 characters, which were received by the interface in "Data Exchange" mode. The characters to be sent are passed in the output bytes.

Table 157: Input/Output Process Image of the Serial Interface, Serial Transmission Mode

| Instance | | Byte designation | Remark |
|---|---|---|---|
| **Input/Output Process Image** | | | |
| | | S0/C0 | Control/status byte S0 |
| | | S1/C1 | Control/status byte S1 |
| | | D0 | Data byte 0 |
| | | D1 | Data byte 1 |
| | | D2 | Data byte 2 |
| | | … | … |
| n | 8 bytes | D5 | Data byte 5 |
| | | D6 | Data byte 6 |
| | | … | … |
| | 24 bytes | D21 | Data byte 21 |
| | | D22 | Data byte 22 |
| | | … | … |
| | 48 bytes | D45 | Data byte 45 |

**Data exchange mode:** The data to be sent and received is stored in up to 47 input and output bytes. The data flow is controlled with the control/status byte.

Table 158: Input/Output Process Image of the Serial Interface, Data Exchange Mode

| Instance | | Byte designation | Remark |
|---|---|---|---|
| **Input/Output Process Image** | | | |
| | | S0/C0 | Control/status byte S0 |
| | | D0 | Data byte 0 |
| | | D1 | Data byte 1 |
| n | | D2 | Data byte 2 |
| | | … | … |
| | 8 bytes | D6 | Data byte 6 |
| | 24 bytes | D7 | Data byte 7 |

Table 158: Input/Output Process Image of the Serial Interface, Data Exchange Mode

| Instance | | Byte designation | Remark |
|---|---|---|---|
| | | … | … |
| | | D22 | Data byte 22 |
| | | D23 | Data byte 23 |
| | | … | … |
| | 48 bytes | D46 | Data byte 46 |

These specialty modules represent 1 x 8, 1 x 24 or 1 x 48 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

### Data exchange module
750-654 (and the versions /000-001)

In the input and output process image, data exchange modules occupy 4 data bytes. 4 bytes are occupied in the process image.

Table 159: Input/Output Process Image, Data Exchange Modules

| Instance | Byte designation | Remark |
|---|---|---|
| n | D0 | |
| | D1 | Data bytes |
| n+1 | D2 | |
| | D3 | |

These specialty modules represent 2 x 2 bytes and occupy 2 instances in class (0x67) and 2 instances in class (0x68).

### WAGO AS-Interface Master
750-655

The image process size for the AS-interface master module is adjustable to 12, 24, 32, 40 or 48 bytes. It consists of a control or status byte, a mailbox with 0, 6, 10, 12 or 18 bytes and 0–32 bytes of AS-interface process data. The AS-interface master module occupies 12–48 bytes in the process image.

The first byte contains the control/status byte, the second contains an empty byte. Subsequently, mailbox data is mapped when the mailbox is permanently superimposed (Mode 1). While in operating mode with a suppressible mailbox (Mode 2), the mailbox and the cyclical process data are mapped next. The remaining bytes contain the remaining process data.

Table 160: Input/Output Process Image, WAGO AS-Interface Master Module

| Instance | Byte designation | Remark |
|---|---|---|
| n | C0/S0 | Control/status byte |
| | - | not used |
| | D0 | Mailbox (0, 6, 12 or 18 bytes) and |

Table 160: Input/Output Process Image, WAGO AS-Interface Master Module

| Input/Output Process Image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| | D1 | process data (0 ... 32 bytes) |
| | D2 | |
| | … | |
| | D46 | |

These specialty modules represent 1 x 12...48 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

## 4-channel IO-Link master
750-657

The length of the process image of the 4-channel IO-Link Master can be adjusted to a fixed size of 4, 6, 8, 10, 12, 16, 20 or 24 bytes.

Table 161: Input/Output Process Image, I/O-Link Master

| Input/Output Process Image | | | | |
|---|---|---|---|---|
| Instance | | Byte designation | Remark | |
| n | | S0/C0 | Control/status byte | |
| | | FC0 | Acyclic channel | Register byte 0 |
| | | MB0 | Mailbox byte | Register byte 1 |
| | 4 bytes | SIO | SIO Byte | |
| | 6 bytes | D0 | Data byte 0 | |
| | | D1 | Data byte 1 | |
| | 8 bytes | D2 | Data byte 2 | |
| | | D3 | Data byte 3 | |
| | 10 bytes | D4 | Data byte 4 | |
| | | D5 | Data byte 5 | |
| | 12 bytes | D6 | Data byte 6 | |
| | | D7 | Data byte 7 | |
| | | D8 | Data byte 8 | |
| | | D9 | Data byte 9 | |
| | | D10 | Data byte 10 | |
| | 16 bytes | D11 | Data byte 11 | |
| | | D12 | Data byte 12 | |
| | | D13 | Data byte 13 | |
| | | D14 | Data byte 14 | |
| | 20 bytes | D15 | Data byte 15 | |
| | | D16 | Data byte 16 | |
| | | D17 | Data byte 17 | |
| | | D18 | Data byte 18 | |
| | 24 bytes | D19 | Data byte 19 | |

These specialty modules represent 1 x 4, 1 x 6 … 1 x 24 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

### CAN gateway
750-658

The length of the process image of the CAN Gateway module can adjusted to a fixed size of 8, 12, 16, 20, 24, 32, 40 or 48 bytes.

### "Sniffer" and "Transparent" Operating Modes

Table 162: CAN Gateway Input/Output Process Image

| Input/Output Process Image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| n | C0/S0 | Control/status byte |
| | MBX0 | |
| | … | 2 … (PI size - 3) |
| | MBXx | |
| | D0 | |
| | … | 5 … 45 data bytes |
| | Dx | |

### "Mapped" Operating Mode

Table 163: CAN Gateway input/output process image

| Input/Output Process Image | | |
|---|---|---|
| Instance | Byte designation | Remark |
| n | C0/S0 | Control/status byte |
| | MBX0 | |
| | … | 2 … (PI size - 3) |
| | MBXx | |
| | T | Toggle byte |
| | D0 | |
| | … | 4 … 44 data bytes |
| | Dx | |

This specialty module represents 1 x 8, 1 x 12 … 1 x 48 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

**Stepper module**
750-670
750-671
750-672
750-673

The stepper module makes a 12-byte input/output process image available.

The data to be sent and received is stored in up to 7 input/output bytes depending on the operating mode. If the mailbox is activated, the first 6 data bytes are overlaid with mailbox data.

Table 164: Input/Output Process Image, Stepper Module

| Input/Output Process Image | | | |
|---|---|---|---|
| **Instance** | **Byte designation** | | **Remark** |
| n | C0/S0 | | Control/status byte |
| | - | | Reserved |
| | D0 | MBX0 | Data bytes / Mailbox bytes (mailbox activated) |
| | D1 | MBX1 | |
| | D2 | MBX2 | |
| | D3 | MBX3 | |
| | D4 | MBX4 | |
| | D5 | MBX5 | |
| | D6 | - | Reserved |
| | C3/S3 | | Status byte (mode specific) |
| | C2/S2 | | Status byte (mode specific) |
| | C1/S1 | | Status byte (mode specific) |

These specialty modules represent 1 x 12 bytes and occupy 1 instance in class (0x67) and 1 instance in class (0x68).

## 12.2.6    System Modules

### System Modules with Diagnostics

750-610, -611

Power supply modules (750-610 and -611) with diagnostics provide 2 bits in the input process image to monitor the power supply.

Table 165: Input Process Image, System Modules 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 Fuse | Diagnostic bit S 1 Voltage |

The system modules occupy 2 instances in class (0x65).

### Binary Spacer Module

750-622

The binary spacer module (750-622) can operate as a 2-channel digital input module or output module and, depending on the setting, and occupies 1, 2, 3 or 4 bits per channel.

Accordingly, 2, 4, 6 or 8 bits are occupied either in the input process image or in the output process image.

Table 166: Input/Output Process Image, Binary Spacer Module

| Input/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 input modules occupy 2, 4, 6 or 8 instances in class (0x65). The output modules occupy 2, 4, 6 or 8 instances in class (0x65).

# 13    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.

# 13.1 Marking Configuration Examples

## 13.1.1 Marking for Europe According to ATEX and IEC-Ex



Figure 67: Side Marking Example for Approved I/O Modules According to ATEX and IECEx



Figure 68: Text Detail – Marking Example for Approved I/O Modules According to ATEX and IECEx.

Table 167: Description of Marking Example for Approved I/O Modules According to ATEX and IECEx

| Printing on Text | Description |
|---|---|
| TÜV 07 ATEX 554086 X<br>IECEx TUN 09.0001 X | Approving authority and certificate numbers |
| **Dust** | |
| II | Equipment group: All except mining |
| 3D | Category 3 (Zone 22) |
| Ex | Explosion protection mark |
| tc Dc | Type of protection and equipment protection level (EPL):protection by enclosure |
| IIIC | Explosion group of dust |
| T 135°C | Max. surface temperature of the enclosure (without a dust layer) |
| **Mining** | |
| I | Equipment group: Mining |
| M2 | Category: High level of protection |
| Ex | Explosion protection mark |
| d Mb | Type of protection and equipment protection level (EPL): Flameproof enclosure |
| I | Explosion group for electrical equipment for mines susceptible to firedamp |
| **Gases** | |
| II | Equipment group: All except mining |
| 3G | Category 3 (Zone 2) |
| Ex | Explosion protection mark |
| nA Gc | Type of protection and equipment protection level (EPL): Non-sparking equipment |
| nC Gc | Type of protection and equipment protection level (EPL): Sparking apparatus with protected contacts. A device which is so constructed that the external atmosphere cannot gain access to the interior |
| IIC | Explosion group of gas and vapours |
| T4 | Temperature class: Max. surface temperature 135°C |
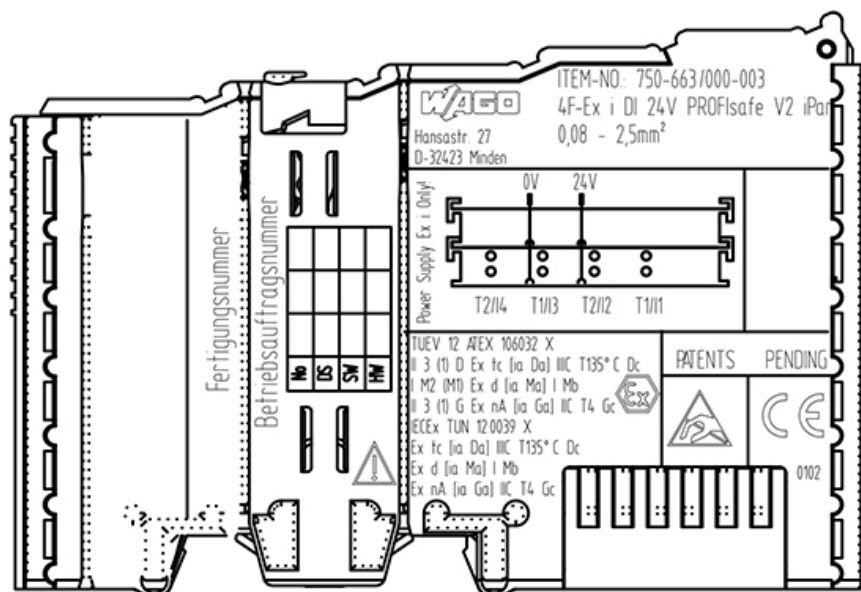
Figure 69: Side Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx.



Figure 70: Text Detail – Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx.

Table 168: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx

| Inscription Text | Description |
|---|---|
| TÜV 07 ATEX 554086 X<br>IECEx TUN 09.0001X<br><br>TÜV 12 ATEX 106032 X<br>IECEx TUN 12.0039 X | Approving authority and certificate numbers |
| **Dust** | |
| II | Equipment group: All except mining |
| 3(1)D | Category 3 (Zone 22) equipment containing a safety device for a category 1 (Zone 20) equipment |
| 3(2)D | Category 3 (Zone 22) equipment containing a safety device for a category 2 (Zone 21) equipment |
| Ex | Explosion protection mark |
| tc Dc | Type of protection and equipment protection level (EPL): protection by enclosure |
| [ia Da] | Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 20 |
| [ib Db] | Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 21 |
| IIIC | Explosion group of dust |
| T 135°C | Max. surface temperature of the enclosure (without a dust layer) |
| **Mining** | |
| I | Equipment Group: Mining |
| M2 (M1) | Category: High level of protection with electrical circuits which present a very high level of protection |
| Ex d Mb | Explosion protection mark with Type of protection and equipment protection level (EPL): Flameproof enclosure |
| [ia Ma] | Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety electrical circuits |
| I | Explosion group for electrical equipment for mines susceptible to firedamp |

Table 168: Description of Marking Example for Approved Ex i I/O Modules According to ATEX and IECEx

| Gases | |
|---|---|
| II | Equipment group: All except mining |
| 3(1)G | Category 3 (Zone 2) equipment containing a safety device for a category 1 (Zone 0) equipment |
| 3(2)G | Category 3 (Zone 2) equipment containing a safety device for a category 2 (Zone 1) equipment |
| Ex | Explosion protection mark |
| nA Gc | Type of protection and equipment protection level (EPL): Non-sparking equipment |
| [ia Ga] | Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 0 |
| [ia Gb] | Type of protection and equipment protection level (EPL): associated apparatus with intrinsic safety circuits for use in Zone 1 |
| IIC | Explosion group of gas and vapours |
| T4 | Temperature class: Max. surface temperature 135°C |

## 13.1.2   Marking for America According to NEC 500
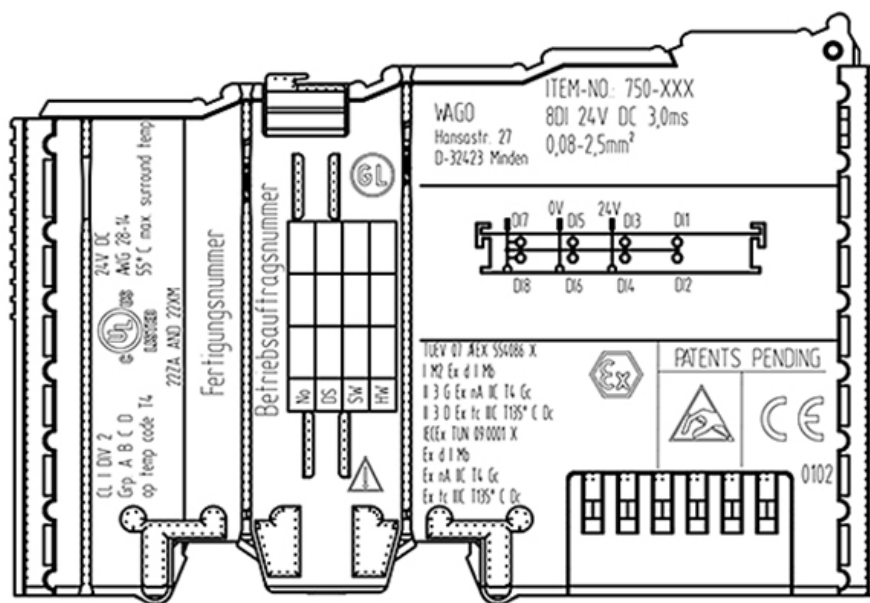


Figure 71: Side Marking Example for I/O Modules According to NEC 500
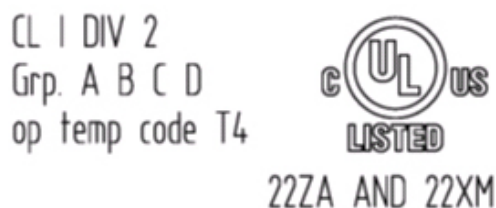


Figure 72: Text Detail – Marking Example for Approved I/O Modules According to NEC 500

Table 169: Description of Marking Example for Approved I/O Modules According to NEC 500

| Printing on Text | Description |
|---|---|
| CL I | Explosion protection group (condition of use category) |
| DIV 2 | Area of application |
| Grp. ABCD | Explosion group (gas group) |
| Op temp code T4 | Temperature class |

## 13.2    Installation Regulations

For the installation and operation of electrical equipment in hazardous areas, the valid national and international rules and regulations which are applicable at the installation location must be carefully followed.

## 13.2.1   Special Conditions for Safe Use (ATEX Certificate TÜV 07 ATEX 554086 X)

1.  For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15 and EN 60079-31.
    For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64.
    The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.
2.  Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3.  Dip-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4.  The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
    This is although and in particular valid for the interfaces "Memory-Card", "USB", "Fieldbus connection", "Configuration and programming interface", "antenna socket", "D-Sub", "DVI-port" and the "Ethernet interface". These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5.  For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.
6.  For replaceable fuses the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7.  The following warnings shall be placed nearby the unit:
    WARNING – DO NOT REMOVE OR REPLACE FUSE WHEN ENERGIZED
    WARNING – DO NOT SEPARATE WHEN ENERGIZED
    WARNING – SEPARATE ONLY IN A NON-HAZARDOUS AREA

## 13.2.2   Special Conditions for Safe Use (ATEX Certificate TÜV 12 ATEX 106032 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** Ex i shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) EN 60079-0, EN 60079-11, EN 60079-15 and EN 60079-31.
   For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to EN 60079-0 and EN 60079-1 and the degree of protection IP64.
   The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExNB.

2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.

3. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.

4. For the type the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in EN 60664-1.

## 13.2.3   Special Conditions for Safe Use (IEC-Ex Certificate TUN 09.0001 X)

1. For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus Independent I/O Modules WAGO-I/O-SYSTEM 750-*** shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15 and IEC 60079-31. For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64.
   The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2. Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3. DIP-switches, binary-switches and potentiometers, connected to the module may only be actuated when explosive atmosphere can be excluded.
4. The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
   This is although and in particular valid for the interfaces "Memory-Card", "USB", "Fieldbus connection", "Configuration and programming interface", "antenna socket", "D-Sub", "DVI-port" and the "Ethernet interface". These interfaces are not energy limited or intrinsically safe circuits. An operating of those circuits is in the behalf of the operator.
5. For the types 750-606, 750-625/000-001, 750-487/003-000, 750-484 and 750-633 the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.
6. For replaceable fuses the following shall be considered: Do not remove or replace the fuse when the apparatus is energized.
7. The following warnings shall be placed nearby the unit:
   WARNING – DO NOT REMOVE OR REPLACE FUSE WHEN ENERGIZED
   WARNING – DO NOT SEPARATE WHEN ENERGIZED
   WARNING – SEPARATE ONLY IN A NON-HAZARDOUS AREA

## 13.2.4    Special Conditions for Safe Use (IEC-Ex Certificate IECEx TUN 12.0039 X)

1.  For use as Gc- or Dc-apparatus (in zone 2 or 22) the Field bus independent I/O Modules WAGO-I/O-SYSTEM 750-*** Ex i shall be erected in an enclosure that fulfils the requirements of the applicable standards (see the marking) IEC 60079-0, IEC 60079-11, IEC 60079-15, IEC 60079-31.
    For use as group I electrical apparatus M2 the apparatus shall be erected in an enclosure that ensures a sufficient protection according to IEC 60079-0 and IEC 60079-1 and the degree of protection IP64.
    The compliance of these requirements and the correct installation into an enclosure or a control cabinet of the devices shall be certified by an ExCB.
2.  Measures have to be taken outside of the device that the rating voltage is not being exceeded of more than 40 % because of transient disturbances.
3.  The connecting and disconnecting of the non-intrinsically safe circuits is only permitted during installation, for maintenance or for repair purposes. The temporal coincidence of explosion hazardous atmosphere and installation, maintenance resp. repair purposes shall be excluded.
4.  For the type the following shall be considered: The Interface circuits shall be limited to overvoltage category I/II/III (non mains/mains circuits) as defined in IEC 60664-1.

## 13.2.5    Special Conditions for Safe Use According to ANSI/ISA 12.12.01

A.    "This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only."

B.    "This equipment is to be fitted within tool-secured enclosures only."

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

D.    "WARNING – Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous" has to be placed near each operator accessible connector and fuse holder.

E.    When a fuse is provided, the following information shall be provided: "A switch suitable for the location where the equipment is installed shall be provided to remove the power from the fuse."

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

G.    "WARNING - Use Module 750-642 only with antenna module 758-910."

H.    For Couplers/Controllers and Economy bus modules only: The instructions shall contain the following: "The configuration interface Service connector is for temporary connection only. Do not connect or disconnect unless the area is known to be non-hazardous. Connection or disconnection in an explosive atmosphere could result in an explosion."

I.    Modules containing fuses only: "WARNING - Devices containing fuses must not be fitted into circuits subject to over loads, e.g. motor circuits."

J.    Modules containing SD card reader sockets only: "WARNING - Do not connect or disconnect SD-Card while circuit is live unless the area is known to be free of ignitable concentrations of flammable gases or vapors."

---

**Information**

**Additional Information**
Proof of certification is available on request.
Also take note of the information given on the operating and assembly instructions.
The manual, containing these special conditions for safe use, must be readily available to the user.

---

# List of Figures

# List of Tables

# WE INNOVATE!

WAGO®