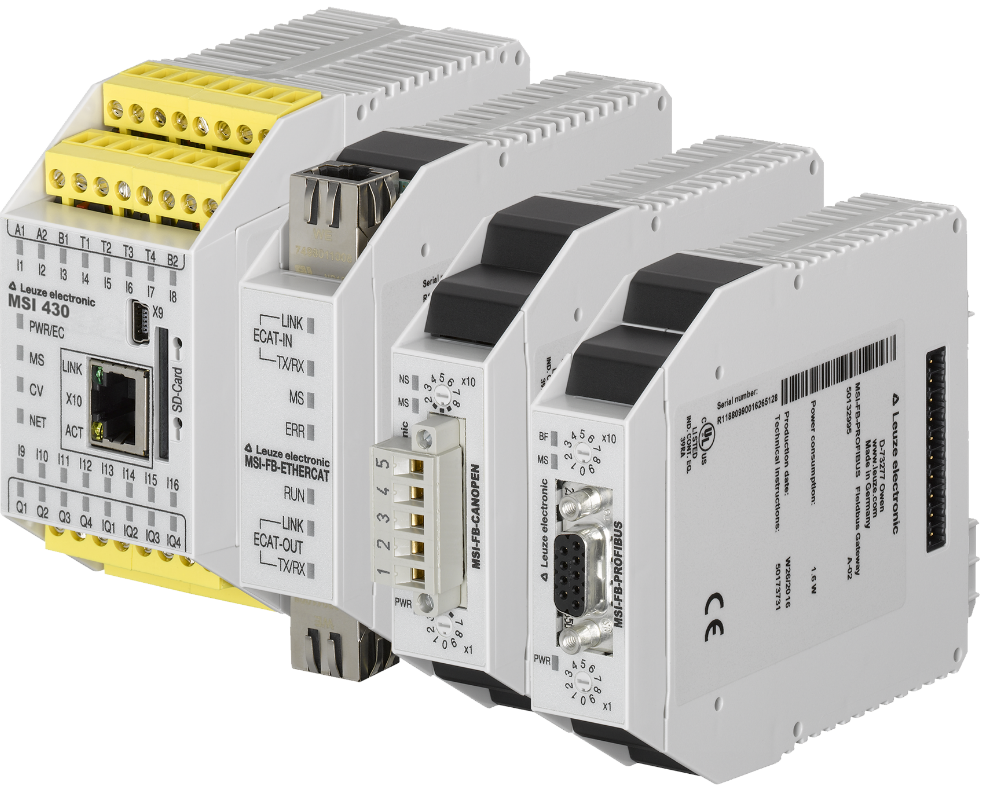
Translation of original operation instruction

# MSI 400 Gateways

Safety controller



The Sensor People

We reserve the right to make technical changes

EN • 1/28/2025 • 50134715-01

Leuze electronic GmbH & Co. KG
In der Braike 1
D-73277 Owen / Germany
Phone: +49 7021 573-0
Fax: +49 7021 573-199
http://www.leuze.com
info@leuze.de

# 1 About this manual

Please read this section carefully before you work with these operating instructions and the MSI 400 gateways.

## 1.1 Function of this document

There are three manuals for the MSI 400 system with clearly delineated areas of application as well as installation instructions and brief instructions for each module.

- This **gateway manual** describes all MSI 400 gateways and their functions in detail. It instructs the technical staff of the machine manufacturer or machine operator in the safe installation, configuration, electrical installation, commissioning, operation and maintenance of the MSI 400 gateways.

  This manual does **not** provide operating instructions for the machine, which incorporates modular MSI 400 safety controllers and a MSI 400 gateway. Information in this regard is provided in the operating instructions for each machine.

  This manual is only valid in combination with the other MSI 400 manuals (see *Scope of validity and applicable documents [chapter 1.2]*).
- The **software manual** describes the software-supported configuration and parameterization of the MSI 400 safety controller. In addition, the software manual contains a description of the important diagnostic functions for operation and detailed information for identifying and eliminating errors. Use the software manual mainly when configuring, commissioning and operating MSI 400 safety controllers.
- The **hardware manual** describes all of the modules and their functions in detail. Use the hardware manual mainly for designing devices.
- Each module contains the **installation instructions/brief instructions**. These instructions provide information on the fundamental technical specifications of the modules and contain simple installation instructions. Use the installation instructions/brief instructions when installing the MSI 400 safety controller.

This manual contains original operating instructions in accordance with the Machinery Directive.

## 1.2 Scope of validity and applicable documents

This manual applies to the following gateway modules:

- MSI-EN-MOD
- MSI-EN-PN
- MSI-EN-IP
- MSI-FB-PROFIBUS
- MSI-FB-CANOPEN
- MSI-FB-ETHERCAT

Tab. 1.1: Overview of the MSI 400 documentation

| Document | Title | Article number |
|---|---|---|
| Software manual | MSI.designer software | 50134712 |
| Hardware manual | MSI 400 hardware | 50134710 |
| Gateway manual | MSI 400 gateways | 50134714 |
| Operating instructions | MSI 400 | 50134613 |
| Operating instructions | MSI-EM-IO84-xx / MSI-EM-I8-xx | 50134614 |
| Operating instructions | MSI-EM-IO84NP-xx | 50134615 |
| Operating instructions | MSI-FB-CANOPEN | 50134616 |
| Operating instructions | MSI-FB-PROFIBUS | 50134617 |
| Operating instructions | MSI-FB-ETHERCAT | 50134618 |

## 1.3 Target group

This manual is intended for **planners**, **developers** and **operators** of systems that incorporate modular MSI 400 safety controllers and that need to exchange data with a field bus (of a control) via a gateway.

It is also aimed at persons commissioning a MSI 400 gateway system for the first time or maintaining such a system.

## 1.4 Information depth

This manual contains information about the following topics related to MSI 400 gateways:

- Mounting
- Integration into the network
- Configuration with the MSI.designer software
- Data transmission to and from the network
- State information, projection and associated mapping
- Item numbers

### Important notes

| ⚠ WARNING |
|---|
| **Observing safety information and protective measures**<br>Observe the safety information and protective measures for the MSI 400 gateways described in this manual. |

### Downloads available from the Internet

Also consult our website on the Internet. At the following link www.leuze.com, you will find:

- the MSI.designer software
- The MSI 400 manuals available for display and printing in various languages:
  - This gateway manual (50134714)
  - The hardware manual (50134710)
  - The software manual (50134712)
- The GSD file of the MSI-FB-PROFIBUS for PROFIBUS-DP
- The EDS file of the MSI-FB-CANOPEN for CANopen

## 1.5 Abbreviations and definitions

| Term | Explanation |
|---|---|
| { } | An element array or an element structure |
| 0b | The following values are specified in binary format |
| 0x | The following values are specified in hexadecimal format |
| Procedure error | A procedure error occurs if, in redundant input circuits, the two input signals are not equal. Monitoring of inequality is frequently carried out within a tolerated time window. |
| ACD | Address Collision Detection |
| ANSI | American National Standards Institute, specified character coding |
| AOI | Add On Instruction |
| AOP | Add On Profile |
| API | Actual Packet Interval |
| AR | Application Relation, unique communication relationship in PROFINET IO between the PLC and the device |
| Attribute | Characteristic or property of an object |

| Term | Explanation |
|---|---|
| Bit | Data unit with a value of 0 or 1 |
| BOOL | Data type specified for CIP devices; stands for a value of 1 byte, in which each of the 8 bits is viewed individually |
| Byte, BYTE | Data unit, representing a sequence of 8 bits; without a plus/minus sign, if not specified |
| CIP | Common Industrial Protocol |
| Controller module | Controller from the MSI 400 product family |
| CRC | Cyclic Redundancy Check, a type or the result of a hash function for revealing errors in the area of data storage or transmission |
| Data block | A data block contains 2-12 bytes of the relevant data set (depending on the gateway used). |
| Data set | Describes a quantity of associated data, e.g. logic values or system state data. A data set can consist of several data blocks. |
| I/O | Input/output |
| EPATH | Encoded Path, especially for CIP applications |
| EtherNet/IP | Industrially-used Ethernet network, combines standard Ethernet technologies with CIP |
| Gateway | Connection module for industrially-used networks, such as EtherNet/IP, PROFIBUS DB, CANopen, Modbus TCP, etc. |
| ID | An identifier or an identity |
| Instance | The physical representation of an object within a class. It stands for one of several objects within the same object class. (Reference: CIP specification, version 3.18) |
| IP | Internet protocol |
| Class | A series of objects representing a similar system component. A class is a generalization of the object, a template for defining variables and methods. All the objects within a class are identical with regard to function and behavior. However, they may have differing attribute values. (Reference: CIP specification, version 3.18) |
| LSB | Low Significant Byte |
| MPI | Measured Packet Interval; shows the API at the time of measurement |
| MSB | Most Significant Byte |
| O→T | Originator to Target (sender to target device) |
| ODVA | Open Device Vendor Association |
| PC | Personal Computer |
| PCCC | Programmable Controller Communication Command |
| PLC | Programmable Logic Controller |
| RPI | Requested Packet Interval |
| RX | Receive |
| S/N | Serial number |
| MSI.designer | Configuration software for controller modules of type MSI 4xx. The software can be run on a PC and communicates with the controller modules. |
| Service | Service to be performed<br>Examples: GetAttributeSingle, SetAttributeSingle |
| SHORT_STRING | Data type specified for CIP devices; stands for a character string (1 byte per character, 1 byte length code) |

| Term | Explanation |
|---|---|
| SINT | Short integer = 1 byte |
| MSI 410<br>MSI 420/430 | Safety controller consisting of a controller module of the MSI 400 product family, as well as optionally connectable expansion gateways and I/O modules. |
| PLC | Programmable Logic Controller<br>( PLC) |
| Stuck-at high | Stuck-at high is an error in which the input or output signal gets stuck at On. The causes for a Stuck-at high can be short-circuits to other input and output lines, often called cross-references, or defective switching elements. Stuck-at-High errors such as sequence errors in dual-channel input circuits are detected using plausibility tests or test pulses in input and output circuits. |
| Stuck-at low | Stuck-at low is an error in which the input or output signal gets stuck at Off. The causes of a stuck-at low can be line interruptions in input circuits or defect switching elements. Stuck-at-Low errors are detected using plausibility tests and do not usually require immediate detection. |
| T→O | Target to Originator |
| TCP | Transmission Control Protocol, Internet standard protocol for the transport layer specified in RFC 793 |
| Test pulses or scan gaps | Test pulses or scan gaps are brief switch-offs / interruptions in input and output circuits, which are generated in a targeted manner to detect stuck-at high errors quickly. Test pulses check the switch-off ability of switching elements during operation on an almost continuous basis. |
| Test pulse error | Test pulse errors are undetected test pulses, which lead to a negative test result and thus switch-off of the affected safety circuits. |
| TX | Transmit / Send |
| UCMM | Unconnected Message Manager |
| UDINT | Unsigned double integer = 4 Bytes = 2 Words<br>Data type specified for CIP applications |
| UDP | User Datagram Protocol, Internet standard protocol for the transport layer specified in RFC 793 |
| UDT | User Defined Type |
| UINT | Unsigned double integer = 2 Bytes = 1 Word<br>Data type specified for CIP applications |
| USINT | Data type specified for CIP applications, which stands for 1 byte without a plus/minus symbol |

## 1.6 Symbols/icons and writing style/spelling standard used

*NOTICE*

These are notes that provide you with information regarding particularities of a device or a software function.

**WARNING**

**Warning!**

A warning lets you know about specific or potential hazards. It is intended to protect you from accidents and help prevent damage to devices and systems.

**Please read and follow the warnings carefully!**
Failure to do so may negatively impact the safety functions and cause a hazardous state to occur.

### Menus and commands

The names of software menus, submenus, options, and commands, selection fields, and windows are written in **bold font**. Example: Click on **Edit** in the **File** menu.

## 1.7 Copyright and right of modification

### Copyright

This document is copyright-protected. The rights derived from this copyright are reserved for Leuze electronic. Reproduction of this document or parts of this document is only permissible within the limits of the statutory provision of the Copyright Act. Any modification or abridgment of the document is prohibited without the express written agreement of Leuze electronic.

Allen-Bradley, CompactBlock Guard I/O, CompactLogix, ControlFLASH, ControlLogix, DH+, FactoryTalk, FLEX, GuardLogix, Kinetix, Logix5000, MicroLogix, PanelBuilder, PanelView, PhaseManager, PLC-2, PLC-3, PLC-5, POINT I/O, POINT Guard I/O, Rockwell Automation, Rockwell Software, RSBizWare, RS-Fieldbus, RSLinx, RSLogix 5000, RSNetWorx, RSView, SLC, SoftLogix, Stratix, Stratix 2000, Stratix 5700, Stratix 6000, Stratix 8000, Stratix 8300, Studio 5000, Studio 5000 Logix Designer, SynchLink, and Ultra are registered trademarks of Rockwell Automation, Inc.

ControlNet, DeviceNet, and EtherNet/IP are registered trademarks of ODVA, Inc.

TwinCAT is a registered trademark of Beckhoff Automation GmbH.

EtherCAT is a registered trademarks and a technology licensed by Beckhoff Automation GmbH.

Microsoft, Windows 10, Windows 11 and .NET Framework are registered trademarks of the Microsoft Corporation.

Any other product or trade names listed in this manual are the trademarks or registered trademarks of the respective owners.

### Subject to change

Subject to technical changes for reasons of continued development.

# 2 Safety

This section is intended to support your safety and the safety of the system users.

↳ Please read this section carefully before you work with a MSI 400 system.

## 2.1 Proper use

The MSI 400 gateways can only be operated in conjunction with a MSI 400 safety controller. The firmware version of the connected controller modules must be at least V1.0.0 and the version of the MSI.designer configuration software must be at least 1.0.0.

### Basic conditions for use

The MSI 400 gateways may only be operated under the following conditions:

- You are operating the gateway within the specified areas of application.
  Further information: Areas of application of the device
- You are operating the gateway within the specified operating limits for voltage, temperature, etc.
  See the following for further information: *Technical data [chapter 12]*
- You are observing personnel requirements.
  Further information: *Qualified persons [chapter 2.3]*
- You are observing the special operator obligations.
  Further information: *Special obligations of the operator [chapter 2.4]*

### Improper use

Any other use or secondary use is deemed improper and is therefore not permitted. Any warranty claims for resulting damage made against Leuze electronic GmbH shall be deemed invalid. The risk shall be borne solely by the operator.

This also applies to any independent modifications made to the device.

## 2.2 Areas of application of the device

### Do not use for safety-related data

⚠ **WARNING**

**Do not operate a MSI 400 gateway on a safety field bus!**

The gateway modules are not suitable for operation with a safety field bus!

They do not only generate safety-related field bus data (state bytes) for control and diagnostic purposes. They do not support any safety mechanisms that would be required for communication within a safety network.

⚠ **WARNING**

**Do not use data from a MSI 400 gateway for safety-related applications!**

The MSI 400 gateways can be used to integrate non-safety-related data into the logic editor in such a way that the safety function of the MSI 400 system may be adversely affected.

↳ Never integrate a gateway into a MSI 400 system without having this source or risk checked by a safety specialist.

### Specifications for domestic use

If you wish to use the MSI 400 system for domestic purposes, you need to take additional steps to prevent the emission of radio frequency interference in limit class B according to EN 55011. Here are some steps you might take:

- The use of interference suppressor filters in the supply circuit
- Installation in grounded switch cabinets or boxes

## 2.3 Qualified persons

A safety controller with MSI 400 gateways may only be installed, commissioned and maintained by qualified persons.

Qualified persons are those who

- have suitable technical training **and**
- have been trained by the machine operator in the operation and applicable safety guidelines **and**
- have access to the MSI 400 system manuals and have read them and duly noted their contents.

## 2.4 Special obligations of the operator

| | ⚠ WARNING |
|---|---|
| ⚠ | **The safety instructions and precautions for use of MSI 400 gateways must be adhered to.**<br>Any other use or any changes to the device – including within the scope of installation – shall nullify any kind of warranty claim against Leuze electronic GmbH. |

**Duty to provide instruction**

- This manual must be made available for the operator of the machine on which the MSI 400 system is to be used. The machine operator must be trained by qualified persons and is required to read this manual.

**Compliance with standards and regulations**

- Please follow the standards and guidelines valid in your country when installing and using the MSI 400 gateways.
- The national/international legal regulations apply to the installation and use of modular MSI 400 safety controllers as well as commissioning and repeated technical testing, particularly the following:
  - EMC Directive 2014/30/EU
  - Work Equipment Directive 2009/104/EC
  - Accident prevention regulations/safety rules

**Requirements for electrical installation**

- The MSI 400 gateways do not have their own power supply.

## 2.5 Environmentally friendly behavior

The Leuze controllers and devices are designed in such a way that they stress the environment as little as possible. They use only a minimum of power and resources.

↳ Make sure that you also carry out work while always considering the environment.

### 2.5.1 Disposal

The disposal of unusable or irreparable devices should always be done in accordance with the respectively valid country-specific waste-elimination guidelines (e.g. European Waste Code 16 02 14).

**NOTICE**

We will be happy to help you in disposing of these devices.
Simply contact us.

### 2.5.2 Sorting of materials

**WARNING**

**Important notes**

↳ The sorting of materials may only be carried out by qualified persons!

↳ Care must be used when disassembling the devices. There is a risk of injuries during this process.

Before you can route the devices to the environmentally-friendly recycling process, it is necessary to sort the various materials of the devices.

↳ Separate the enclosure from the rest of the components (particularly from the PC board).

↳ Place the separated components into the corresponding recycling containers (see the following table).

Tab. 2.1: Overview of disposal according to components

| Components | Disposal |
|---|---|
| Product<br>Housing<br>PC boards, cables, connectors, and electric connecting pieces | <br>Aluminum recycling<br>Plastic recycling<br>Electronics recycling |
| Packaging<br>Cardboard, paper | <br>Paper/cardboard recycling |

# 3 Product description

MSI 400 gateways allow a MSI 400 system to transmit non-safety-related data for control and diagnostic purposes to the external field bus system and to receive them.

**Important safety information**

| | ⚠ WARNING |
|---|---|
| ⚠ | **Do not operate a MSI 400 gateway on a safety field bus!**<br>The gateway modules are not suitable for operation with a safety field bus!<br>They do not only generate safety-related field bus data (state bytes) for control and diagnostic purposes. They do not support any safety mechanisms that would be required for communication within a safety network. |

**Information on the function, configuration and designations**

| | *NOTICE* |
|---|---|
| ! | Where not otherwise indicated, this manual always considers the data exchanged between the MSI 400 system and the relevant network from the point of view of the network master (PLC). Thus data sent to the network from the MSI 400 system is termed input data, while data received from the network is termed output data. |

Configuration of MSI 400 gateways takes place via the MSI.designer configuration software, using a PC or Notebook connected to the MSI 4xx main module via the USB interface or RJ45 Ethernet interface.

The safety-related logic of the MSI 400 system works independently of the gateway. However, if the system has been configured in such a way that non-safety-related information from the field bus can be integrated into the logic editor, switching off the gateway may result in availability problems.

A MSI 400 gateway can only be operated on a MSI 400 system. It does not have its own power supply. A maximum of two MSI 400 gateways can be operated simultaneously for each system.

Order information: *Order data [chapter 13]*

## 3.1 Version, compatibility, and features

There are various module versions and function packages for the MSI 400 product family that enable various functions. This section will give you an overview as to which module version you will need to be able to use a certain function or a certain device.

Tab. 3.1: Versions of the gateway modules

| Gateway | Module name | Version |
|---|---|---|
| EtherCAT | MSI-FB-ETHERCAT | A-02 |
| PROFIBUS DP | MSI-FB-PROFIBUS | A-03 |
| CANopen | MSI-FB-CANOPEN | A-02 |

**Info**

- You can find the module version on the type plate of the modules.
- You will find the MSI.designer version in the main menu.
- The latest software version is available in the Internet at the following address www.leuze.com.
- Newer modules are backwards-compatible, which means that each module can be replaced with a module having a higher module version.
- You can find the date of manufacture for a device on the type plate in the **S/N** field in the format <Product no.>yywwnnnnn (yy = year, ww = calendar week).

## 3.2 Equipment variants

There are six MSI 400 gateways for various network types.

The Modbus TCP / PROFINET IO, EtherNet/IP or MSI-FB-ETHERCAT gateway is suitable for Ethernet networks. The MSI-FB-PROFIBUS and MSI-FB-CANOPEN gateway are used for fieldbus communication.

**WARNING**

**Restrictions for Ethernet connections**

- The Ethernet connection can only be linked to autonomous networks or demilitarized zones (DMZ).
- The device must never be connected directly to the Internet.
- Always use secure data tunnels (VPN) to exchange data via the Internet.

Tab. 3.2: Equipment variants and their main characteristics

| Gateway | Network type | Ethernet IP socket interface |
|---|---|---|
| MSI-EN-MOD | Modbus TCP with master and slave operation | Client/Server on TCP Port 502 |
| MSI-EN-PN | PROFINET IO device | UDP ports 34964, 49152 |
| MSI-EN-IP | EtherNet/IP device | TCP port 44818, UDP port 2222 |
| MSI-FB-PROFIBUS | PROFIBUS DP slave | - - |
| MSI-FB-CANOPEN | CANopen slave | - - |
| MSI-FB-ETHERCAT | EtherCAT slave | - - |

***NOTICE***

You will find the manufacturing date of a device on the type label in the S/N field in the format yywwnnnn (yy = year, ww = calendar week, nnnn = sequential serial number within a calendar week).

## 3.3 Data transferred to the network (network input data sets)

### Available data

The MSI 400 gateways can provide the following data:

- Process data
  - **Logic results from** the MSI 400 safety controller (see *Routing table [chapter 5.1.3]*)
  - **Input values** (HIGH/LOW) for all MSI 400 input expansion modules in the system
  - **Output values** (HIGH/LOW) for all MSI 400 input/output expansion modules (see *Module state / input and output values [chapter 3.3.1]*)
  - **Output data** from another network, i.e. data received from a second gateway in the MSI 400 system (see *Transmission of data from a second network [chapter 3.3.3]*)
- Diagnostics
  - **Test values** (CRCs): (see *Data set 2 [chapter 11.5.2]*)
  - **Error and state information**: *Error and state information for the modules [chapter 3.3.4]*

### Default values

Data from gateways or the non-secure MSI-EM-IO84NP I/O expansion module is categorically not secure. Default values are taken up in the error state of the controller. The default value of input/output values is 0 and the default value of status values is 1.

### Data sets

The physical modules are not presented as typical hardware modules in the network. Instead, the data provided by the MSI 400 system has been arranged in four *input data sets*.

- **Data set 1** (max. 50 bytes) contains the process data. It can be compiled with the aid of MSI.designer. In the form in which it is delivered, the content of data set 1 is preconfigured; it can be freely modified. Details: see table *"Overview of input data sets" [chapter 3.3]*

  For the MSI-FB-PROFIBUS gateway, data set 1 was divided into five input data blocks, with data blocks 1–4 each containing 12 bytes and data block 5 two bytes.
  For the MSI-FB-CANOPEN gateway, data set 1 was divided into four blocks, each with 8 bytes.
  You will find more detailed information in the corresponding section for each gateway.
- **Data set 2** (32 bytes) contains the test values (CRCs) for the system configuration.

  See table “Overview of input data sets 1–3 (basic settings for Modbus TCP)" below
- **Data set 3** (60 bytes) contains the state and diagnostic data for the various modules, with four (4) bytes per module, with the controller module comprising 3 x 4 bytes. Details: see table *"Meaning of module state bits" [chapter 3.3.4]*
- **Data set 4** (60 bytes) is currently filled with reserved values.

**Note**: As of build state MSI 400, the setting of multibit values (16 bit and 32 bit) is supported. The format and dimensioning is described in the software manual (see Mapping function block values to gateways).

The following table provides an overview of which data sets are provided by which gateway.

Tab. 3.3: Availability of data sets 1–4

| | Data set 1 | Data set 2 | Data set 3 | Data set 4 |
|---|---|---|---|---|
| MSI 430-x<br>???- LEUZE | Modbus TCP<br>PROFINET IO<br>EtherNet/IP | Modbus TCP<br>PROFINET IO<br>EtherNet/IP | Modbus TCP<br>PROFINET IO<br>EtherNet/IP | Modbus TCP<br>PROFINET IO |
| MSI-FB-ETHER-CAT | EtherCAT | EtherCAT | EtherCAT | - |
| MSI-FB-PROFIBUS | PROFIBUS DP | – | – | – |
| MSI-FB-CANOPEN | CANopen | SDOs[1] | SDOs[1] | – |

[1] The MSI-FB-CANOPEN is used to provide diagnostic data via CANopen SDO (service data objects). More information about how to provide state and diagnostic data with the aid of the CANopen gateway may be found here: *CANopen gateway [chapter 10]*

[2] Readable with instance 2 of class 120

[3] Readable with instance 3 of class 120 and byte 52 to 111 of assembly 167

Tab. 3.4: Overview of input data sets 1–3 (basic setting for Modbus TCP)

| | Data set 1 | Data set 2 | Data set 3 | Data set 4 |
|---|---|---|---|---|
| Byte 0 | Input values for Module 0 (I1..I8) | Project CRC | Module state MSI 4xx | Reserved |
| Byte 1 | Input values for Module 0 (I9..I16) | | Module state MSI 4xx | |
| Byte 2 | Input values for Module 0 (IQ1..IQ4) | | Test pulse comparison, controller module inputs | |
| Byte 3 | Output values for Module 0 (Q1..Q4, IQ1..IQ4) | | Test pulse comparison, controller module inputs | |
| Byte 4 | Direct data (Off) 0 | Internal CRC[1] | Test pulse comparison, controller module inputs | |
| Byte 5 | Direct data (Off) 1 | | State of two-channel controller module inputs | |
| Byte 6 | Direct data (Off) 2 | | State of two-channel controller module inputs | |
| Byte 7 | Direct data (Off) 3 | | Reserved | |

| | Data set 1 | Data set 2 | Data set 3 | Data set 4 |
|---|---|---|---|---|
| Byte 8 | Direct data (Off) 4 | Reserved | Stuck-at error at controller module outputs | Reserved |
| Byte 9 | Direct data (Off) 5 | | Stuck-at error at controller module outputs | |
| Byte 10 | Direct data (Off) 6 | | Reserved | |
| Byte 11 | Direct data (Off) 7 | | Reserved | |
| Byte 12 | Input values for Module 1 | | State of Module 1 | |
| Byte 13 | Input values for Module 2 | | State of Module 1 | |
| Byte 14 | Input values for Module 3 | | State of Module 1 | |
| Byte 15 | Input values for Module 4 | | State of Module 1 | |
| Byte 16 | Input values for Module 5 | | State of Module 2 | |
| Byte 17 | Input values for Module 6 | | State of Module 2 | |
| Byte 18 | Input values for Module 7 | | State of Module 2 | |
| Byte 19 | Input values for Module 8 | | State of Module 2 | |
| Byte 20 | Input values for Module 9 | | State of Module 3 | Reserved |
| Byte 21 | Input values for Module 10 | | State of Module 3 | |
| Byte 22 | Input values for Module 11 | | State of Module 3 | |
| Byte 23 | Input values for Module 12 | | State of Module 3 | |
| Byte 24 | Output values for Module 1 | Reserved | State of Module 4 | Reserved |
| Byte 25 | Output values for Module 2 | | State of Module 4 | |
| Byte 26 | Output values for Module 3 | | State of Module 4 | |
| Byte 27 | Output values for Module 4 | | State of Module 4 | |
| Byte 28 | Output values for Module 5 | | State of Module 5 | |
| Byte 29 | Output values for Module 6 | | State of Module 5 | |
| Byte 30 | Output values for Module 7 | | State of Module 5 | |
| Byte 31 | Output values for Module 8 | | State of Module 5 | |
| Byte 32 | Output values for Module 9 | Not available | State of Module 6 | Reserved |
| Byte 33 | Output values for Module 10 | | State of Module 6 | |
| Byte 34 | Output values for Module 11 | | State of Module 6 | |
| Byte 35 | Output values for Module 12 | | State of Module 6 | |
| Byte 36<br>…<br>Byte 47 | Not allocated | | State of Module 7<br>…<br>Status of Module 9 | |
| Byte 48 | | | State of Module 10 | |
| Byte 49 | | | State of Module 10 | |

| | Data set 1 | Data set 2 | Data set 3 | Data set 4 |
|---|---|---|---|---|
| Byte 50 | Not available | Not available | State of Module 10 | Reserved |
| Byte 51 | | | State of Module 10 | |
| Byte 52<br>…<br>Byte 55 | | | State of Module 11<br>…<br>Status of Module 11 | |
| Byte 56 | | | State of Module 12 | |
| Byte 57 | | | State of Module 12 | |
| Byte 58 | | | State of Module 12 | |
| Byte 59 | | | State of Module 12 | |
| Length | 50 bytes | 32 bytes | 60 bytes | 60 bytes |
| [1] The use of the internal CRC in data set 2 is only permitted for diagnostic purposes so that Leuze Technical Support can continue to provide support | | | | |

***NOTICE***

When dual-channel input or output elements have been configured for an I/O module, only the lowest bit constitutes the input or output state (on/off) of the corresponding element. It is represented by the tag name of the element. The highest bit represents the state of this input/output.

***NOTICE***

The input values in data set 1 do not represent the physical state at the input terminals, but the pre-processed input values that are used for logic processing.

### 3.3.1 Direct gateway output values

It is possible to write values directly from the **Logic** view to a gateway. Four bytes have been reserved for this purpose in the basic settings for data set 1; however, up to the total number of 50 bytes of data set 1 may be configured as direct gateway output values. You can obtain additional information at: *Direct gateway output values [chapter 5.3]*.

***NOTICE***

Configured bytes must be specified in the gateway view so that they can be used in the "Logic Editor".

### 3.3.2 Module state / input and output values

The MSI 400 gateways can transmit the input and output states of all modules connected to the MSI 400 system to the network. Data set 3 contains a non-modifiable configuration. Moreover, data set 1 can be adapted to contain up to 4 bytes of collective state information. Only the input and output values for data set 1 have been predefined and these can be freely adapted. You will find more detailed information in the section on the relevant gateway, as well as in the following section: *Configuration of gateways with MSI.designer [chapter 5]*

**Module state**

The MSI 400 gateways can transfer the state of the linked modules to the network. A total of 4 bytes are available for this purpose.

Tab. 3.5: Module state

| Module state | Size | Meaning | Assignment |
|---|---|---|---|
| **Input data state** | 2 bytes | One sum bit per module for the state of the module inputs<br>0 = error 1 = no error | Bit 0 = MSI 4xx<br>Bit 1 = 1.<br>Extension module<br>Bit 2 = 2nd<br>Expansion module … Bit 13 = 1.<br>Gateway Bit 14 = 2. Gateway Bit 15 = reserved |
| **Output data state** | 2 bytes | One sum bit per module for the state of the module outputs<br>0 = error 1 = no error | |

You will find information about the meaning of the state bits at: software manual, Internal inputs for controller modules

**Input and output values for the modules**

- **Input values for I/O modules**
  1 byte for data set 1 is available for every expansion module. The input values show the state of the preliminary evaluation of the I/O module. This corresponds to the state of the element in the controller module logic. The level at the associated terminal cannot be clearly detected from this, as the data may be set to low, irrespectively of the level at the input terminal, by means of the cross-connection detection or two-channel evaluation (e.g. I1-18).
  When two-channel input elements have been configured for an I/O module, only the lower-value bit represents the pre-evaluation state of the corresponding element (e.g. bit 0 for I1 and I2, bit 2 for I3 and I4, bit 4 for I5 and I6, bit 6 for I7 and I8). The higher-value bit (bit 1, 3, 5 and 7) is used as follows in this case:
  0 = error 1 = no error
- **Output values for I/O modules**
  1 byte for data set 1 is available for every module with outputs. The output values indicate the state of the control information from the logic of the controller module for the relevant element of the I/O module. The level of the associated terminals cannot be clearly detected from this, as the output may be switched off via the cross-connection detection or the overload connection function.
  When two-channel output elements have been configured for an I/O module, only the lower-value bit represents the control information (e.g. bit 0 for Q1 and Q2, bit 2 for Q3 and Q4, bit 4 for Q5 and Q6, bit 6 for Q7 and Q8). The higher-value bit (bit 1, 3, 5 and 7) is not used as follows in this case (low):

### 3.3.3 Transmission of data from a second network

If your MSI 400 system contains two gateways, it is possible to forward information which the first gateway receives from a network (e.g. from a Modbus PLC) via the second gateway to a second network (e.g. to a PROFIBUS master) and vice versa.

### 3.3.4 Error and state information for the modules

Data set 3 and 4 contain the state information for the modules that will be transferred to the network.

Ten bytes are transmitted for MSI 4xx controller module. For each MSI-EM-I8 and MSI-EM-IO84 I/O module, four bytes are transmitted in Little Endian format, e.g. as a 32-bit word, with the first byte being placed in the least significant byte of the whole number (far right) and the fourth byte in the most significant byte of the whole number (far left).

Data sets 3 and 4 cannot be adapted.

**Module status bits of the controller module MSI 4xx**

The module state bits have the following meaning, if not otherwise indicated:

0 = error

1 = no error

Reserved bits have the value 1

**NOTICE**

You can find an explanation of the technical terms used below here: *Abbreviations and definitions [chapter 1.5]*

Tab. 3.6: Meaning of the module status bits of the MSI 430-x/???- LEUZE controller module

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Byte 0** | B2 status | Collective error fast shut-off | B1 status | Configuration state | A1 status | External module state | Internal module state | Reserved |
| **Byte 1** | Module state output data | Module state of input data | Reserved | Reserved | IQ3+IQ4 power requirement 0: Overcurrent 1: no overcurrent | IQ1+IQ2 power requirement 0: Overcurrent 1: no overcurrent | Q3+Q4 power requirement 0: Overcurrent 1: no overcurrent | Q1+Q2 power requirement 0: Overcurrent 1: no overcurrent |
| **Byte 2** | I8 vs. T2/4 test pulse comparison | I7 vs. T1/3 test pulse comparison | I6 vs. T2/4 test pulse comparison | I5 vs. T1/3 test pulse comparison | I4 vs. T2/4 test pulse comparison | I3 vs. T1/3 test pulse comparison | I2 vs. T2/4 test pulse comparison | I1 vs. T1/3 test pulse comparison |
| **Byte 3** | I16 vs. T2/4 test pulse comparison or HW limit frequency I16 | I15 vs. T1/3 test pulse comparison or HW limit frequency I15 | I14 vs. T2/4 test pulse comparison or HW limit frequency I14 | I13 vs. T1/3 test pulse comparison or HW limit frequency I13 | I12 vs. T2/4 test pulse comparison | I11 vs. T1/3 test pulse comparison | I10 vs. T2/4 test pulse comparison | I9 vs. T1/3 test pulse comparison |
| **Byte 4** | 0: Cable break at I16<br>1: OK or not used | 0: Cable break at I15<br>1: OK or not used | 0: Cable break at I14<br>1: OK or not used | 0: Cable break at I13<br>1: OK or not used | IQ4 vs. T2/4 test pulse comparison | IQ3 vs. T1/3 test pulse comparison | IQ2 vs. T2/4 test pulse comparison | IQ1 vs. T1/3 test pulse comparison |
| **Byte 5** | I15/I16 dual-channel status 0: Error 1: ok or not used | I13/I14 dual-channel state 0: Error 1: ok or not used | I11/I12 dual-channel state 0: Error 1: ok or not used | I9/I10 dual-channel state 0: Error 1: ok or not used | I7/I8 dual-channel state 0: Error 1: ok or not used | I5/I6 dual-channel state 0: Error 1: ok or not used | I3/I4 dual-channel state 0: Error 1: ok or not used | I1/I2 dual-channel state 0: Error 1: ok or not used |

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Byte 6** | 0: Inversion window of Sensor 2<br>1: OK or not used | 0: Inversion window of Sensor 1<br>1: OK or not used | 0: Frequency difference I14 vs. I16<br>1: OK or not used | 0: Frequency difference I13 vs. I15<br>1: OK or not used | 0: Phase difference I14 vs. I16 too low<br>1: OK or not used | 0: Phase difference I13 vs. I15 too low<br>1: OK or not used | IQ3/IQ4 dual-channel state<br>0: Error<br>1: ok or not used | IQ1/IQ2 dual-channel state<br>0: Error<br>1: ok or not used |
| **Byte 7** | 0: I16 Stuck-at low<br>1: OK or not used | 0: I16 Stuck-at high<br>1: OK or not used | 0: I15 Stuck-at low<br>1: OK or not used | 0: I15 Stuck-at high<br>1: OK or not used | 0: I14 Stuck-at low<br>1: OK or not used | 0: I14 Stuck-at high<br>1: OK or not used | 0: I13 Stuck-at low<br>1: OK or not used | 0: I13 Stuck-at high<br>1: OK or not used |
| **Byte 8** | Q4 Stuck-at low | Q4 Stuck-at high | Q3 Stuck-at low | Q3 Stuck-at high | Q2 Stuck-at low | Q2 Stuck-at high | Q1 Stuck-at low | Q1 Stuck-at high |
| **Byte 9** | IQ4 (Output) Stuck-at low | IQ4 (Output) Stuck-at high | IQ3 (Output) Stuck-at low | IQ3 (Output) Stuck-at high | IQ2 (Output) Stuck-at low | IQ2 (Output) Stuck-at high | IQ1 (Output) Stuck-at low | IQ1 (Output) Stuck-at high |

**Module state bits of the I/O modules MSI-EM-I8 and MSI-EM-IO84**

The module state bits have the following meaning, if not otherwise indicated:

0 = error

1 = no error

Tab. 3.7: Meaning of the module state bits of the safe I/O modules MSI-EM-I8 and MSI-EM-IO84

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Byte 0** | Reserved | Collective error fast shut-off | Power supply for Q1 .. Q4 | Configuration of this module is valid. | Not used (error history flag) | External module state | Internal module state | Not used ("executing state") |
| **Byte 1** | Module state of output data | Module state of input data | Reserved | Reserved | Two-channel evaluation of input I7–I8 | Two-channel evaluation of input I5–I6 | Two-channel evaluation of input I3–I4 | Two-channel evaluation of input I1–I2 |
| **Byte 2** | Test impulse comparison I8 vs. X2 | Test impulse comparison I7 vs. X1 | Test impulse comparison I6 vs. X2 | Test impulse comparison I5 vs. X1 | Test impulse comparison I4 vs. X2 | Test impulse comparison I3 vs. X1 | Test impulse comparison I2 vs. X2 | Test impulse comparison I1 vs. X1 |
| **Byte 3** | Q4 Stuck-at low<br>0: Stuck-at error<br>1: no stuck-at | Q4 Stuck-at high<br>0: Stuck-at error<br>1: no stuck-at | Q3 Stuck-at low<br>0: Stuck-at error<br>1: no stuck-at | Q3 Stuck-at high<br>0: Stuck-at error<br>1: no stuck-at | Q2 Stuck-at low<br>0: Stuck-at error<br>1: no stuck-at | Q2 Stuck-at high<br>0: Stuck-at error<br>1: no stuck-at | Q1 Stuck-at low<br>0: Stuck-at error<br>1: no stuck-at | Q1 Stuck-at high<br>0: Stuck-at error<br>1: no stuck-at |

### Module state bits of the MSI-EM-IO84NP I/O module

The module state bits have the following meaning if not otherwise indicated; normally only the first byte of the total state is transmitted:

0 = error

1 = no error or reserved

Tab. 3.8: Meaning of the module state bits of the MSI-EM-IO84NP expansion module

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Byte 0** | Reserved | Reserved | Power supply Y1-Y4 and IY5-IY8 | Configuration status | Not used (error history flag) | External module state | Internal module state | Not used ("executing state") |
| **Byte 1** | Module state output data | Module state input data | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| **Byte 2** | Reserved | | | | | | | |
| **Byte 3** | Reserved | | | | | | | |

### Module state bit of the gateways

The module state bits have the following meaning if not otherwise indicated; normally only the first byte of the total state is transmitted:

0 = error

1 = no error

Tab. 3.9: Meaning of gateway module state bits

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **Byte 0** | Reserved | Module state output data | Module state input data | Configuration status | Not used (error history flag) | Reserved | Internal module state | Not used ("executing state") |
| **Byte 1** | Reserved | | | | | | | |
| **Byte 2** | Reserved | | | | | | | |
| **Byte 3** | Reserved | | | | | | | |

### Example

Module 2 (MSI-EM-IO84) has a short-circuit after high (24 V) at output 3. The following module state is transmitted to the network (only the first 20 of 60 bytes are shown):

| Byte address | 00 | 01 | 02 … | 03 11 | 04 12 | 05 13 | 06 14 | 07 15 | 08 16 | 09 17 | 10 18 | 11 19 | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Byte** | 3 0 | 2 1 | 1 … | 0 11 | 3 0 | 2 1 | 1 2 | 0 3 | 3 0 | 2 1 | 1 2 | 0 3 | … |
| **Value** | FF | FF | FF | FF | FF | FF | FF | FF | **EF FB** | FF | FF | **FB EF** | … |
| **Meaning** | Controller module status | | | | Module 1 status (MSI-EM-IO84) | | | | **State of module 2 (**MSI-EM-IO84**)** | | | | … |

The first relevant byte for the module 2 error described above is module state byte 0 for module 2. This is byte 11 with the hexadecimal value FB (1111 1011):

| **Bit #** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| **Value** | 1 | 1 | 1 | 1 | 1 | **0** | 1 | 1 |

This corresponds to the error message "Summary of bits 0.5 to 0.7 (external error)", byte 0, bit 2 in the following table: *"Meaning of module state bits of the secure I/O modules" [chapter 3.3.4]*

The second relevant byte is the module state byte 3 for module 2. This is byte 08 with the hexadecimal value EF (1110 1111):

| **Bit #** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| **Value** | 1 | 1 | 1 | **0** | 1 | 1 | 1 | 1 |

This corresponds to the error message "Short circuit monitoring of output 3, short circuit after high", byte 3, bit 4 in the following table: *"Meaning of module state bits of the secure I/O modules" [chapter 3.3.4]*

**NOTICE**

- Reserved (for future use) = static 1 (no state change)
- Not used (can be 0 or 1 ), both values occur.
- If there is no module, all values - including the reserved values - are set to logical 1.

### 3.3.5 Transmission time of input and output data via an external gateway

The transmission time and thus the delay of the data depends on the number and amount of configured gateway data.

1 gateway and up to 10 bytes of data 4 ms additional delay.

2 gateways and up to 10 bytes of data 8 ms additional delay.

1 gateway and up to 50 bytes of data 20 ms additional delay.

2 gateways and up to 50 bytes of data 40 ms additional delay.

**NOTICE**

For every 10 bytes, there is a delay of 4 ms. The maximum amount of configured data in one direction is received and a second gateway doubles this time.

## 3.4 Data received from the network (network output data sets)

The data from data set 1 (max. 50 bytes) received from the network may be differently arranged, depending on the protocol. For the Modbus TCP, this data set was divided into five data blocks, each with 10 bytes. In the MSI-FB-PROFIBUS gateway, output data blocks 1-4 each contain 12 bytes, while output data block 5 contains 2 bytes. CANopen only defines 4 data blocks, each with 8 bytes.

Tab. 3.10: Output data block 1–5 of the various gateways

| | **Size of output data block** | | | | |
|---|---|---|---|---|---|
| **Gateway** | **Block 1** | **Block 2** | **Block 3** | **Block 4** | **Block 5** |
| MSI-FB-PROFIBUS / PROFINET IO | 12 bytes | 12 bytes | 12 bytes | 12 bytes | 2 bytes |
| MSI-FB-CANOPEN | 8 bytes | 8 bytes | 8 bytes | 8 bytes | – |
| MSI-FB-ETHERCAT / Modbus TCP / EtherNet/IP | 10 bytes | 10 bytes | 10 bytes | 10 bytes | 10 bytes |

The content of the output data blocks can be used in the logic editor, as well as made available for another network via a second gateway within the MSI 400 system.

**NOTICE**

- In order to use network data in the logic editor or as input for another network, you must assign a tag name for each bit to be used.
- Bits without specific tag names will not be available in the logic editor or for routing via a second gateway. Detailed information about how to assign tag names for the data received may be found in the corresponding sections of the chapters on the various gateways.
- You can monitor current communication with the network with the aid of input data state bits for receiving data from the network and the output data state bit for transmitting data to the network in the logic editor. When the gateway detects a communication error, both the content of the data sets and the associated state bit are set to zero (logical 0).
- When all communication fails, the data of the output data sets and the input data state bit are set to zero (logical 0).
- When a connection is closed while others remain available, the LED MS or LED state will flash red/green for a total of 10 seconds and an entry will be made in the error log. In this case the state bits are not affected.

**WARNING**

**Do not use the same output data block number for two different PLC connections or TCP/IP sockets!**

The output data block of the Ethernet gateways can be described in parallel via all communication interfaces or TCP/IP sockets (e.g. Modbus TCP/IP and Ethernet TCP/IP) if they make use of the same output data block number. In this case the last message will always overwrite the data received earlier.

# 4 Installation and basic configuration

## 4.1 Installing/removing

### 4.1.1 Installing modules on DIN rails

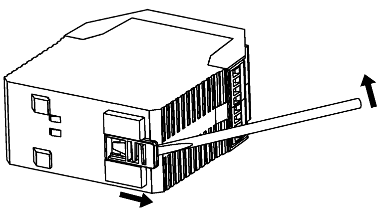
| ⚠ WARNING | |
|---|---|
| ⚠ | **Only for control cabinets with protection class IP 54 or higher!**<br>The MSI 400 system is only suitable for installations in a switchbox having at least protection class IP 54. |

**Notes**

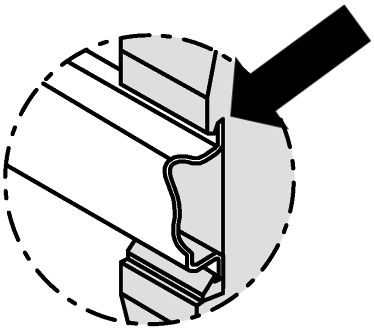
- Basic safety:
  Gateways and extended modules may not be removed or added when the operating voltage is switched on.
- Grounding:
  The DIN rail must be conductively connected to the protective earth conductor (PE).
- ESD protection measures:
  Observe suitable ESD protection measures during installation.
  Failure to do so could result in damage to the modules.
- Protect connector openings:
  Undertake suitable measures so that no foreign bodies can penetrate connector openings, particularly those for the program removable storage.
- Module width:
  The modules are placed in a mounting box that is 22.5 mm or 45 mm wide depending on type.
- Quality of DIN rail:
  The mounting boxes are suitable for 35 mm DIN rails as per EN 60715.
- Sequence of modules:
  The MSI 400 system has the controller module on the far left. The two optional gateways follow directly to the right next to the controller module. The expansion modules only follow thereafter.
  As a general principle, we recommend that a distance of ≥ 15 mm is provided between the last system module and the adjacent modules on the right. This measure makes module replacement easier and prevents interference from possible malfunctions in the module extension connector.
- Standards to be taken into consideration:
  Installation according to EN 50274

**Step 1: Installing a controller module**

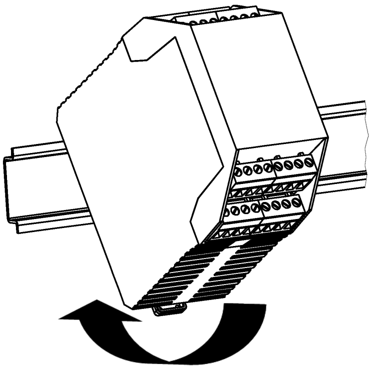
↳ Use a screwdriver to pull the mounting foot outward.



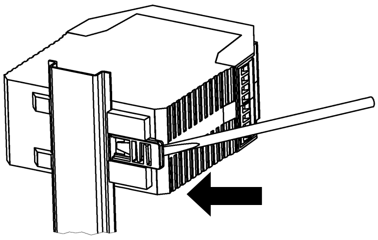
↳ Hang the module on the DIN rail.
**Important!** Make sure that the screening spring fits correctly.
The screening spring of the module must be placed onto the DIN Rail so that it is secure and has good electrical contact.



↳ Fold the module onto the DIN rail.



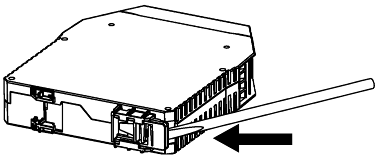
↳ Use a screwdriver to move the mounting foot against the DIN rail until the mounting foot latches into position with an audible click.



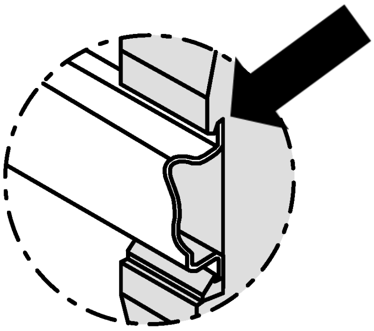
↳ Make sure that the module is securely seated on the DIN rail.
Attempt to pull the module from the DIN rail using slight pressure. If the module stays connected to the DIN rail during this test, then the installation is correct.

**Step 2: Installation of gateways or expansion modules**

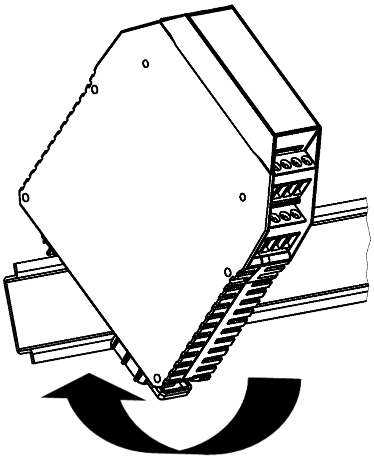
↳ Use a screwdriver to pull the mounting foot outward.



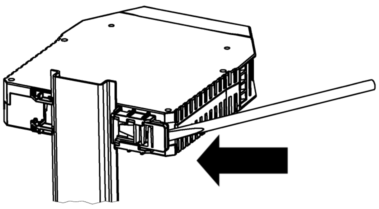
↳ Hang the module on the DIN rail.
**Important!** Make sure that the screening spring fits correctly.
The screening spring of the module must be placed onto the DIN Rail so that it is secure and has good electrical contact.



↳ Fold the module onto the DIN rail.

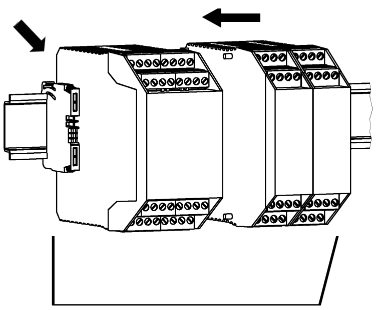


↳ Use a screwdriver to move the mounting foot against the DIN rail until the mounting foot latches into position with an audible click.



↳ Make sure that the module is securely seated on the DIN rail.
Attempt to pull the module from the DIN rail using slight pressure. If the module stays connected to the DIN rail during this test, then the installation is correct.

↳ If you are installing multiple modules:
Push the individual modules together in the direction of the arrow until the lateral plug connection between the modules audibly latches into position.



↳ Install an end terminal into the module furthest to the left and another end terminal into the module furthest to the right.

**After installation**

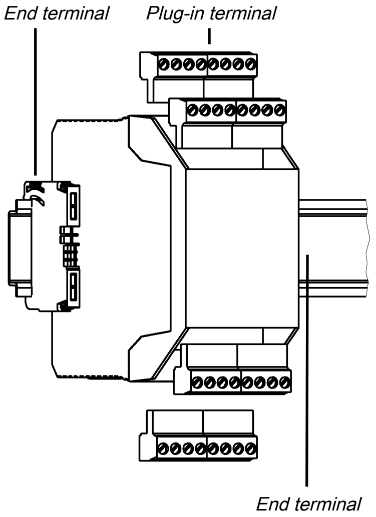
Once you have installed the modules, the following steps are required:

- Connect the modules electrically.
- Configure modules (see: software manual).
- Check the installation before first commissioning.

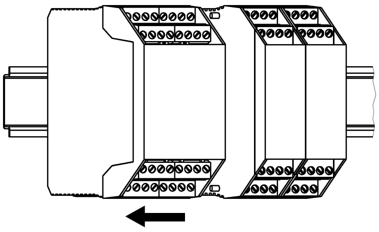
### 4.1.2 Removing modules from the DIN rail

**Step 1: Removing a controller module**

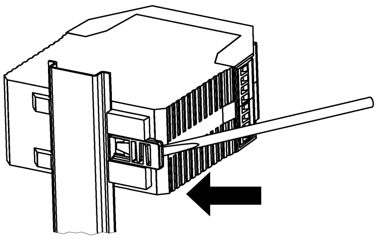
- ↳ De-energize the MSI 400 system.
- ↳ Remove plug-in terminals with wiring and remove the end terminal.



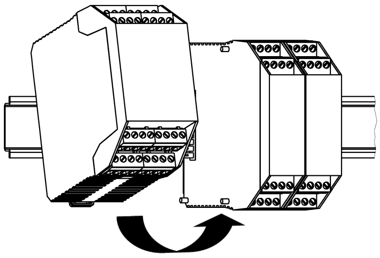

- ↳ If expansion modules or gateways are used:
  Slide the controller module in the direction of the arrow until the lateral plug connection is disconnected.



- ↳ Unlock the module.
  To do this, pull the mounting foot of the module outward using a screwdriver.

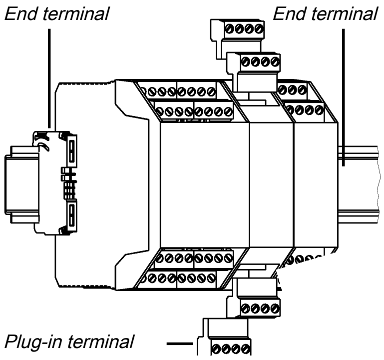


- ↳ Fold the module away from the DIN rail and remove from the rail.

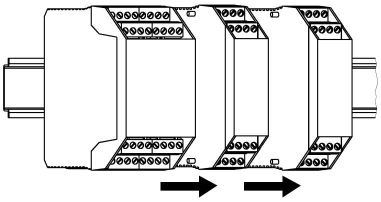


**Step 2: Removing gateways and expansion modules**

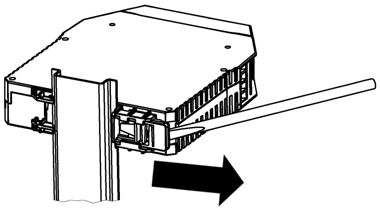
- ↳ De-energize the MSI 400 system.
- ↳ Remove any plug-in terminals with wiring and remove the end terminals.



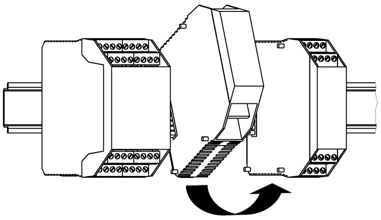

↳ Pull the modules apart from one another individually in the direction of the arrow until the lateral plug connection is disconnected.



↳ Unlock the module.
To do this, pull the mounting foot of the module outward using a screwdriver.



↳ Fold the module away from the DIN rail and remove from the rail.



## 4.2 Electrical installation

**WARNING**

**Switch off the power supply to the system!**

It is possible for the system to be unexpectedly started while you are connecting the devices.

***NOTICE***

- MSI 400 gateways meet EMC conditions as set out in the EN 61000-6-2 specification for use in an industrial environment.
- To ensure complete EMC safety, the DIN rail must be connected to the functional earth (FE).
- The switch box or installation housing for the MSI 400 system must meet at least the requirements of protection class IP 54.
- Installation according to EN 50274.
- Electrical installation as per EN 60204-1.
- The external power supply of the devices must be able to bridge a short-term power outage of 20 ms in accordance with EN 60204-1.
- The power supply must meet the provisions for low-voltage with safe disconnection (SELV, PELV) in accordance with EN 60664-1.
- Ensure that all modules of the MSI 400 system, the connected protective devices and the power supplies are connected to the same ground connection. The ground of the RS-232 interface is internally connected to the ground of the power supply for the controller module (A2).
- Connect the screening of all field bus and Ethernet cables to functional earth (FE) just before they lead into the control cabinet.

## 4.3 Initial configuration steps

How do you configure gateways? This chapter provides some brief guidelines.

Tab. 4.1: Guidelines for gateway configuration

| Step | Description |
|---|---|
| 1 | **Establishing a link between the gateway and PC**<br>See here for more detailed information:<br>Software manual, chapter "Connecting to the safety controller" |
| 2 | **Configure gateway**<br>You will find detailed information in this regard at the following points in the gateway manual:<br>• *Modbus TCP gateway [chapter 6]*<br>• *PROFINET IO-Gateway [chapter 7]*<br>• *EtherNet/IP gateway [chapter 8]*<br>• *PROFIBUS DP gateway [chapter 9]*<br>• *CANopen gateway [chapter 10]*<br>• *EtherCAT gateway [chapter 11]* |
| 3 | **Transmitting and verifying the configuration**<br>See here for more detailed information:<br>Software manual, chapter "Transferring the system configuration" |

# 5 Configuration of gateways with MSI.designer

This chapter gives you an overview of how to configure gateways in MSI.designer. It explains

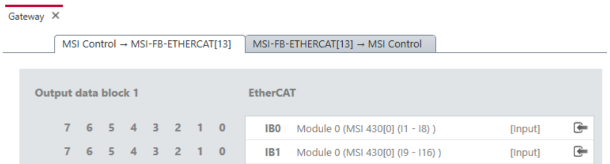
- how the graphical user interface is laid out for the gateway configuration in MSI.designer,
- how you can carry out typical configuration tasks connected to gateways in MSI.designer.

**NOTICE**

You will find more detailed information about the graphical user interface of MSI.designer in the Software manual.

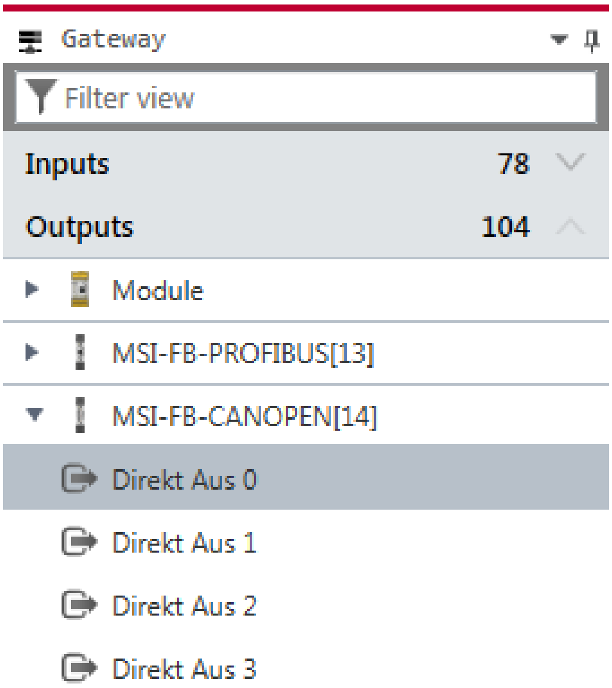
## 5.1 The graphical user interface

You can edit the configuration for gateways in the graphical user interface of MSI.designer in the following windows:




Depending on module configuration, shows two or three tabs with the routing tables and additional gateway functionalities.

Details: *Layout and content of the tabs [chapter 5.1.3]*




Show inputs and outputs available for the gateway configuration as a hierarchical tree structure.

Details: *"Gateway" and "Properties" docking windows [chapter 5.1.4]*

### 5.1.1 Activating gateway functionality

The **Gateway** view is only available in MSI.designer when you actively use the gateway functionality. Basically you can set up the gateway functionality in two ways:

**Scenario 1: You are using a gateway module**

You implement the gateway functionality via a supplementary module, as indicated in the following example. Here the MSI-FB-CANOPEN module is used:

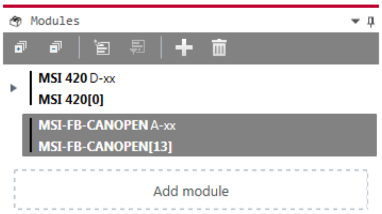



Fig. 5.1: Module configuration with gateway module

**Scenario 2: You are using the gateway function on the** MSI 430-x/???- LEUZE module.

You implement the gateway functionality via the controller module. In this case, you must use a MSI 430-x/???- LEUZE module as the controller module for the controller and explicitly set the gateway functionality there.

This is how to activate the gateway function on the MSI 430-x/???- LEUZE module:

- ✎ Open the **Modules** docking window.
- ✎ Select the MSI 430-x/???- LEUZE module.


- ✎ Open the **Properties** docking window.
- ✎ Select the desired gateway function from the **Gateway** selection list.

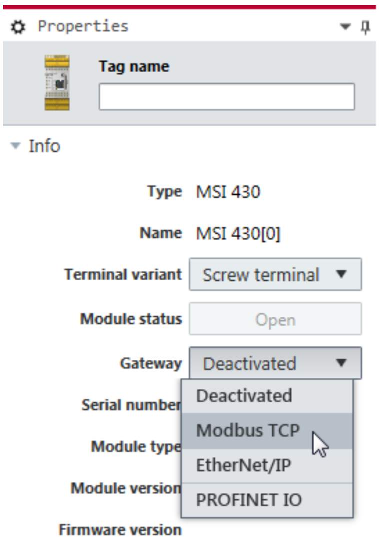


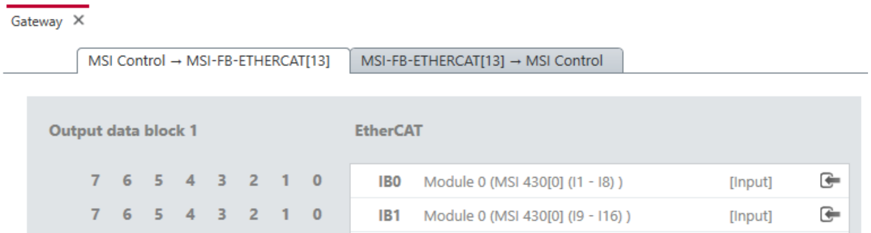
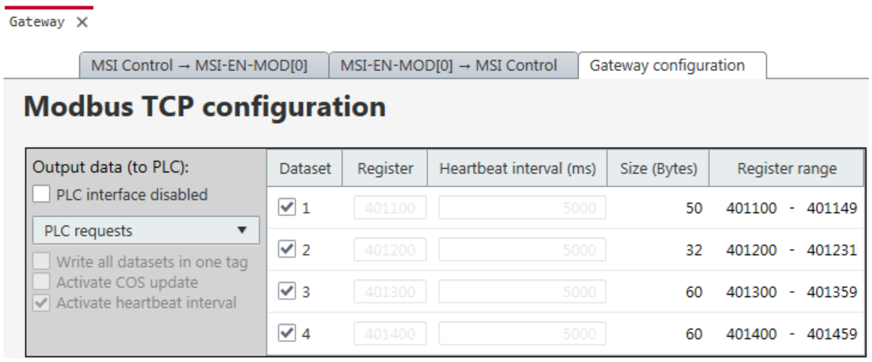

Fig. 5.2: MSI 430-x/???- LEUZE module with activated gateway function

### 5.1.2 "Gateway" view

If you have activated the gateway functionality in MSI.designer automatically the **Gateway** view is active. There you can edit the gateway configuration.

**Design**

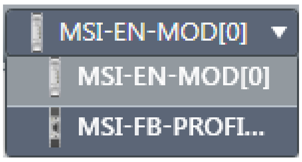
Depending on module configuration, in the **Gateway** view, you will see two or three tabs:

| Module configuration | Design |
|---|---|
| You use a gateway module for gateway configuration | In the **Gateway** view you see two tabs with the routing tables for the input and output values.<br>Example: MSI-FB-ETHERCAT module<br><br> |
| You are using the gateway function of the MSI 430-x/???-LEUZE module. | In the **Gateway** view you see an additional third tab **Gateway configuration**.<br>This contains the basic settings for the gateway function of the MSI 430-x/???- LEUZE module:<br><br> |

**Visualization**

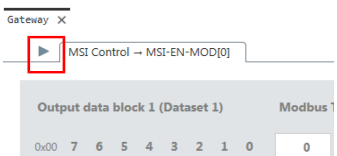
- **If you are using several gateways**

  The **Gateway** view always only shows one gateway configuration. If you are using several gateways, you can toggle between the configurations by making use of the **Select data set view** menu:

- **When the program window is very small**

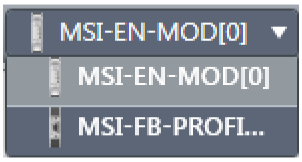
  If the window in which you have opened MSI.designer is very small, not all tabs may be shown. In this case an arrow symbol will appear, allowing you to toggle between the tabs:

**Commands**

Via the command bar of the **Gateway** view, you have access to the following view-specific features:

Tab. 5.1: Key

| Element | Description |
|---|---|
|  | Only with a connection to the controller:<br>Stops the controller. |
|  | Only with a connection to the controller:<br>Starts a stopped controller. |
|  | **Zoom**<br>This determines the size of the display in the **Gateway** view work area. |
|  | **Undo**<br>This renders the last action undone. |
|  | **Redo**<br>This makes an action that has been undone redone. |
|  | **Standard**<br>This resets the configuration of the gateways to the basic settings.<br>Also see: *Basic settings for the process data [chapter 5.2.2]* |
|  | **Data set view selection**<br>When you are using several gateways: Changes between the gateway configurations. |
|  | **Importing/exporting**<br>Allows for the import/export of the configuration defined in the **Gateway** view.<br>**Notes**:<br>• **Important:** When you import a configuration, all changes made before that have not been saved will be lost. You cannot undo this command.<br>• Available storage formats: SPG, XML, CSV<br>You can use the import/export function to import the tag names used for a project into a PLC program, or to export them from a PLC program into MSI.designer. |
|  | **Delete**<br>This deletes the currently selected element. |

### 5.1.3 Layout and content of the tabs

The tabs of the **Gateway** view contain the following data and features:

#### Tab 1: Routing table with output values (data bytes)

Transmission direction: MSI 400 -> Network/field bus

The mapping is shown in tabular form. Bits which have been used appear on a dark blue background. In online mode, the input data of the relevant gateway is displayed (byte display 0x00 at the start of the relevant line).

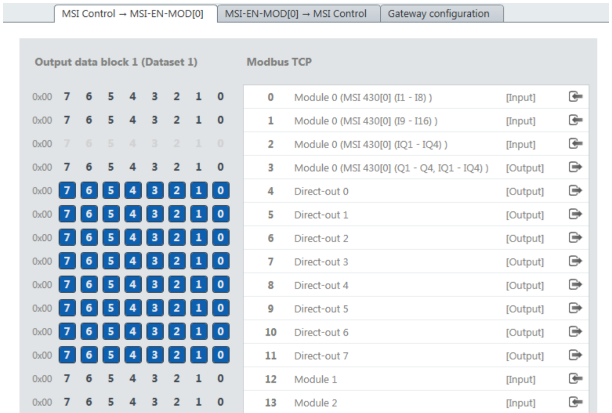



Fig. 5.3: Routing table with output values

#### Tab 2: Routing table with input values (data bytes)

Transmission direction: Network/field bus -> MSI.designer

Visualization: as per Tab 1

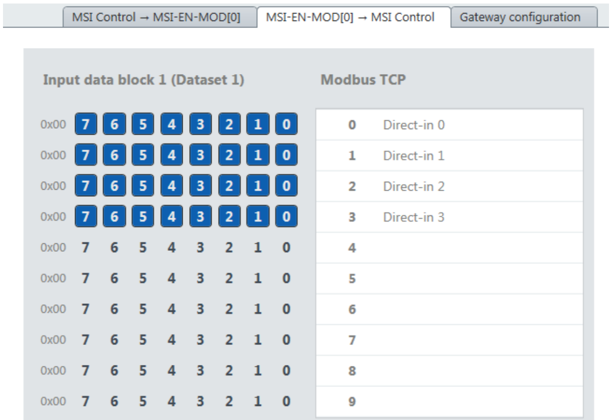



Fig. 5.4: Routing table with input values

#### Tab 3: "Gateway configuration"

Tab 3 only appears if you have activated MSI-EN-MOD or MSI-EN-IP.

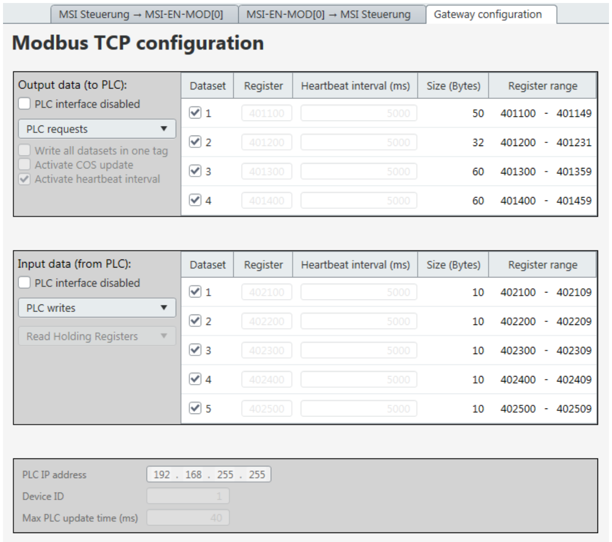



Fig. 5.5: "Gateway configuration" tab for MSI-EN-MOD

| | **NOTICE** |
|---|---|
| | **Allocation of input and output data**<br>The output and input data listed here refer directly to the data blocks in tab 1 and tab 2.<br>↳ **Output data (to PLC)** group:<br>Only **data set 1** can be configured. This refers directly to **Output data block 1** in tab 1.<br>↳ **Input data** group **(from PLC):**<br>**Data set 1** to **data set 5** refers directly to **Input data block 1** to **Input data block 5** in tab 2. |

### 5.1.4 "Gateway" and "Properties" docking windows

In addition to the **Gateway** view, in the gateway configuration in MSI.designer, you work with the following windows:

#### "Gateway" docking window

From the **Gateway** docking window you can drag hardware data bytes into empty cells in the routing table (**Gateway** view).

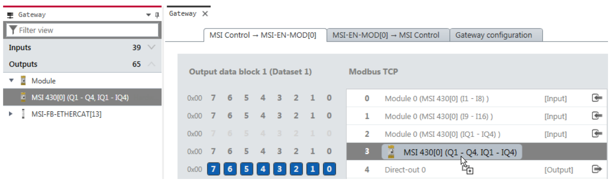



Fig. 5.6: Drag data bytes into the routing table using the drag-and-drop function

| | **NOTICE** |
|---|---|
| | You can make use of the same data byte several times in the routing table. |

#### "Properties" docking window

In the **Properties** docking window, the configuration dialog appears for the data byte which you have selected in the **Gateway** view.

Depending on the data byte, you can configure individual parameters. You can also allocate tag names here.

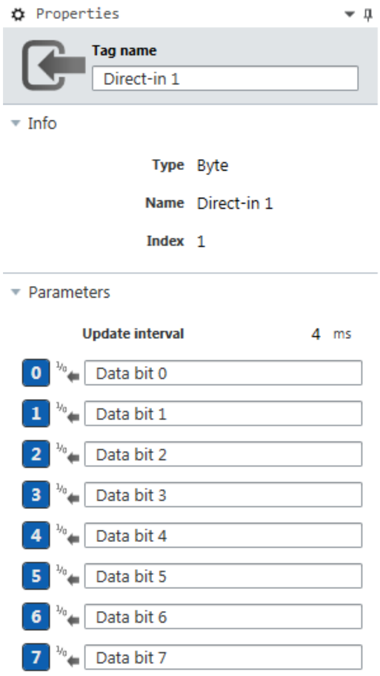



Fig. 5.7: Configuration dialog in the “Properties” docking window

## 5.2 Function and basic settings

### 5.2.1 Routing

The process diagram, transferred to the network from the MSI 400 gateway, comprises the operating data (e.g. logic results, state of inputs and outputs) and the diagnostic data (e.g. module status, CRCs). This data have been arranged in 4 data sets.

Tab. 5.2: Content of data sets 1–4

| Data set | Content | Size | Configurable |
|---|---|---|---|
| 1 | Process data | 50 bytes | Yes |
| 2 | CRCs | 32 bytes | No |
| 3 | State and diagnosis | 60 bytes | No |
| 4 | Reserved | 60 bytes | No |

The process data in Data Set 1 may consist of up to 50 bytes, irrespective of the network protocol used. These 50 bytes have been divided into one or several data blocks, depending on the network protocol. Detailed information about the modularization of the data sent to the network may be found in the section on the relevant gateway and in the following table: *"Preset configuration for process data transmitted in the network" [chapter 5.2.2]*

The content of data set 1 has been pre-configured with the addition of a gateway module or a gateway function, but can be freely configured with a granularity of 1 byte (see *Basic settings for process data [chapter 5.2.2]* and *Configuring the gateway output values (tab 1) [chapter 5.3]*).

The diagnostic data in data sets 2-4 depends on the network protocol used and is described in the chapter on the relevant gateway.

### 5.2.2 Basic settings for the process data

After the addition of the gateway, the process data is pre-configured. Depending on the gateway used, this data is divided into several data blocks.

The following table provides an overview of which bytes have been allocated to the preset configuration and how the data at the various gateways are modularized.

Tab. 5.3: Preset configuration for the process data transmitted in the network

| | Modbus TCP | | PROFIBUS DP | |
|---|---|---|---|---|
| **Byte** | **Preset allocation** | **Initial data set** | **Preset allocation** | **Initial data block** |
| 0 | Input values for Module 0 (I1..I8) | #1 (50 bytes) | Input values for Module 0 (I1..I8) | #1 (12 bytes) |
| 1 | Input values for module 0 (I9..I16) [1)] | | Input values for Module 0 (I9..I16) | |
| 2 | Input values for Module 0 (IQ1..IQ4) | | Input values for Module 0 (IQ1..IQ4) | |
| 3 | Output values for Module 0 (Q1..Q4,IQ1-IQ4) | | Output values for Module 0 (Q1..Q4,IQ1-IQ4) | |
| 4 | Direct data (Off) 0 | | Direct data (Off) 0 | |
| 5 | Direct data (Off) 1 | | Direct data (Off) 1 | |
| 6 | Direct data (Off) 2 | | Direct data (Off) 2 | |
| 7 | Direct data (Off) 3 | | Direct data (Off) 3 | |
| 8 | Direct data (Off) 4 | | Direct data (Off) 4 | |
| 9 | Direct data (Off) 5 | | Direct data (Off) 5 | |
| 10 | Direct data (Off) 6 | | Direct data (Off) 6 | |
| 11 | Direct data (Off) 7 | | Direct data (Off) 7 | |
| 12 | Inputs for Module 1 | *Continued* #1 (50 bytes) | Inputs for Module 1 | #2 (12 bytes) |
| 13 | Inputs for Module 2 | | Inputs for Module 2 | |
| 14 | Inputs for Module 3 | | Inputs for Module 3 | |
| 15 | Inputs for Module 4 | | Inputs for Module 4 | |
| 16 | Inputs for Module 5 | | Inputs for Module 5 | |
| 17 | Inputs for Module 6 | | Inputs for Module 6 | |
| 18 | Inputs for Module 7 | | Inputs for Module 7 | |
| 19 | Inputs for Module 8 | | Inputs for Module 8 | |
| 20 | Inputs for Module 9 | | Inputs for Module 9 | |
| 21 | Inputs for Module 10 | | Inputs for Module 10 | |
| 22 | Inputs for Module 11 | | Inputs for Module 11 | |
| 23 | Inputs for Module 12 | | Inputs for Module 12 | |

| | Modbus TCP | | PROFIBUS DP | |
|---|---|---|---|---|
| **Byte** | **Preset allocation** | **Initial data set** | **Preset allocation** | **Initial data block** |
| 24 | Outputs for Module 1 | *Continued*<br>#1<br>(50 bytes) | Outputs for Module 1 | #3<br>(12 bytes) |
| 25 | Outputs for Module 2 | | Outputs for Module 2 | |
| 26 | Outputs for Module 3 | | Outputs for Module 3 | |
| 27 | Outputs for Module 4 | | Outputs for Module 4 | |
| 28 | Outputs for Module 5 | | Outputs for Module 5 | |
| 29 | Outputs for Module 6 | | Outputs for Module 6 | |
| 30 | Outputs for Module 7 | | Outputs for Module 7 | |
| 31 | Outputs for Module 8 | | Outputs for Module 8 | |
| 32 | Outputs for Module 9 | | Outputs for Module 9 | |
| 33 | Outputs for Module 10 | | Outputs for Module 10 | |
| 34 | Outputs for Module 11 | | Outputs for Module 11 | |
| 35 | Outputs for Module 12 | | Outputs for Module 12 | |
| 36-47 | Not allocated | *Continued*<br>#1<br>(50 bytes) | Not allocated | #4<br>(12 bytes) |
| 48-49 | Not allocated | | Not allocated | #5<br>(2 bytes) |

[1)] Due to the predominantly high dynamics of the individual signal values at I13-I16, no individual bits are mapped into these input values in motion monitoring sensors. This also applies to all other gateway protocols.

The preset allocation of the bytes can be freely configured, as shown in the following section.

## 5.3 Configuring the gateway output values (tab 1)

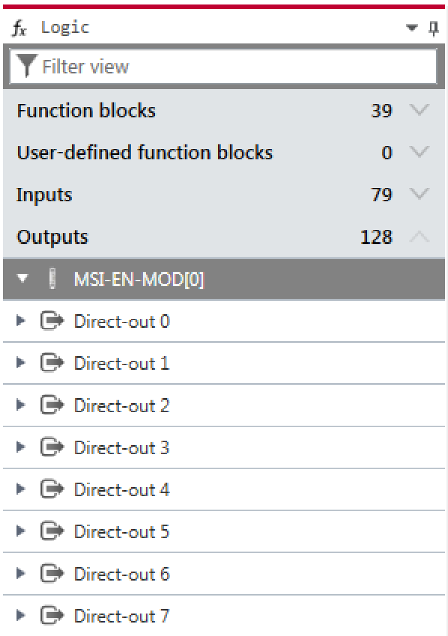
You can use the following settings for the output values of a gateway in tab 1:

### Basic setting

Depending on the gateway function selected, you will find four or eight bytes in tab 1, which are reserved as direct gateway output values. You can also see these bytes in the **Logic** docking window.

| | ***NOTICE*** |
|---|---|
| ! | Configured bytes must be specified in the gateway view so that they can be used in the "Logic Editor". |

Example: MSI-FB-CANOPEN module with eight predefined outputs for gateways:

| "Logic" docking window | "Gateway" view |
|---|---|
| In the **Logic** docking window you see these four bytes **Outputs**: | In the **Gateway** view these four outputs appear in the first tab: |
|  |  |





### Change tag names of a predefined output value

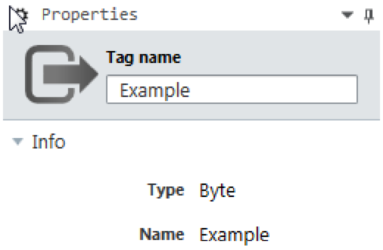
Tag names have already been pre-assigned to the predefined output values (bytes). You can change these tag names:

↳ In the **Gateway** view click on the byte, whose tag name you want to change.

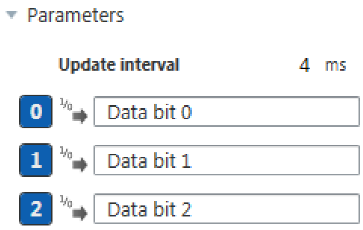


↳ Open the **Properties** docking window.

↳ If you wish to change the tag name of the byte:
Overwrite the pre-allocated tag name of the byte with the desired new value in the configuration dialog.




- If you also want to change the tag names of individual bits:
  Overwrite the pre-allocated values with the desired new value under **Parameters** in the configuration dialog.

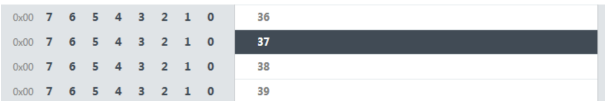



In the **Logic** view, these bits will appear with the corresponding tag names.

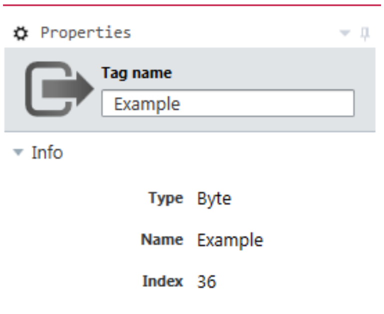
**Configuring additional direct gateway output values**

You can add new output values (bytes) in addition to the pre-allocated output values in the **Gateway** view.

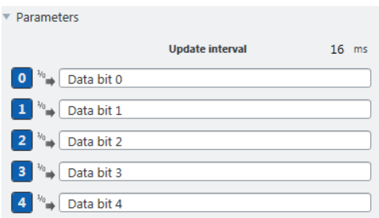
- Click on an empty byte in the Gateway view.



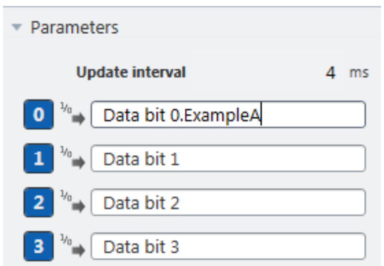

- Open the **Properties** docking window and assign a tag name for the byte in the configuration dialog.




⇨ Tag names for all bits are automatically pre-allocated under **Parameters**.




- If you want to change the tag names of individual bits:
  Overwrite the pre-allocated values with the desired new value under **Parameters** in the configuration dialog.



In the **Logic** view, these bits will appear with the corresponding tag names.

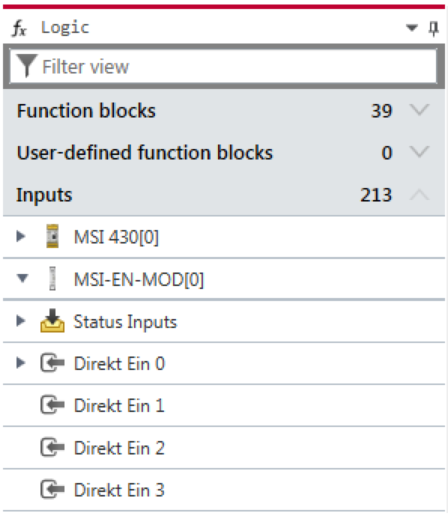
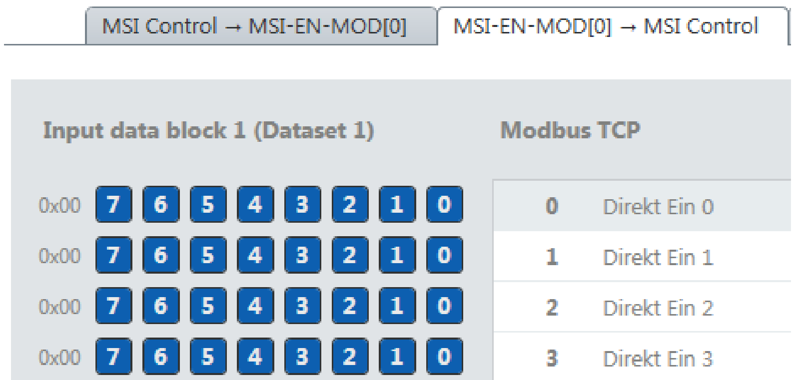
## 5.4 Editing the gateway input values (tab 2)

You can use the following settings for the output values of a gateway in tab 2:

### Basic setting

Depending on the gateway function selected, you will find four or eight bytes in tab 2, which are reserved as direct gateway input values. You can also see these bytes in the **Logic** docking window.

Example: MSI-FB-CANOPEN module with four predefined inputs for gateways:

| "Logic" docking window | "Gateway" view |
|---|---|
| In the **Logic** docking window you see these four bytes under **Inputs**: | In the **Gateway** view, these four inputs appear in tab 2: |
|  |  |

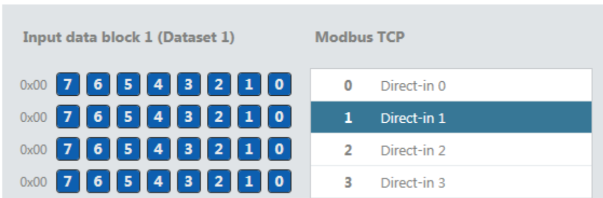




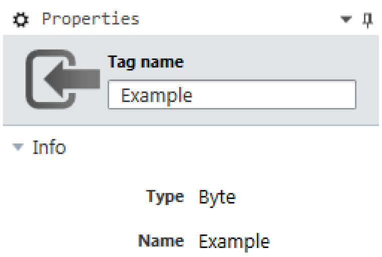
### Change tag names of a predefined input value

Tag names have already been pre-assigned to the predefined input values (bytes). You can change these tag names:

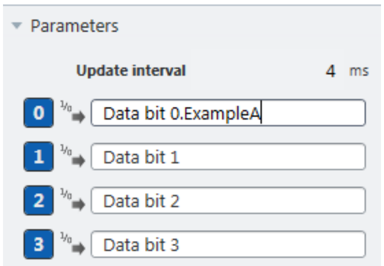
- ✎ In the **Gateway** view click on the byte, whose tag name you want to change.




- ✎ Open the **Properties** docking window.
- ✎ If you wish to change the tag name of the byte:
  Overwrite the pre-allocated tag name of the byte with the desired new value in the configuration dialog.




- ✎ If you also want to change the tag names of individual bits:
  Overwrite the pre-allocated values with the desired new value under **Parameters** in the configuration dialog.



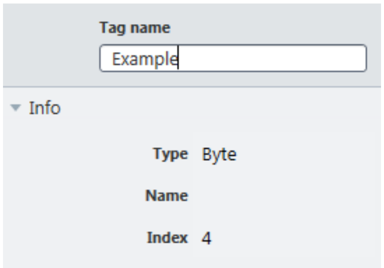

In the **Logic** view, these bits will appear with the corresponding tag names.

### Configuring additional gateway input values

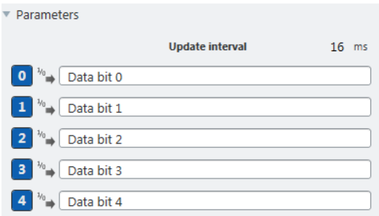
You can add new output values (bytes) in addition to the pre-allocated output values in the **Gateway** view.

↳ Click on an empty byte in the Gateway view.

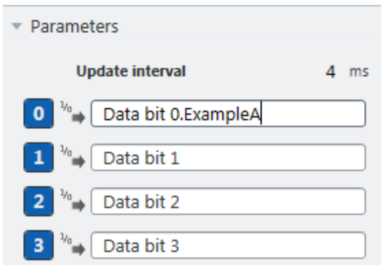

↳ Open the **Properties** docking window and assign a tag name for the byte.




⇨ Tag names for all bits are automatically pre-allocated under Parameters.




↳ If you also want to change the tag names of individual bits:
Overwrite the pre-allocated values with the desired new value under **Parameters**.




In the **Logic** view, these bits will appear with the corresponding tag names.

## 5.5 Monitoring process data

You can monitor your gateway configuration directly in MSI.designer. This can be done in simulation mode (limited monitoring option) or by means of an active link to a MSI 400 system.

**NOTICE**

The MSI 400 gateways always show the actual physical state of the inputs and outputs of the connected modules and equipment. This means that even when the force mode is active and inputs that are physically **Low** are forced to **High** (or vice versa), the actual physical state of these inputs is transmitted to the PLC and not the (virtual) forced state. However, if one or several outputs change their state as a result of one or several inputs being forced, the changed state of these outputs will also be transmitted to the PLC, as the actual physical state of the equipment outputs has changed.

### Simulation mode (offline mode)

You can test a gateway configuration offline in simulation mode. Use the logic analyzer for this purpose and manually set the desired inputs to **High** or **Low**.

Read here how to work with the simulation mode and logic analyzer: Software manual, chapter "Simulating logic programming"

### Monitoring with an active connection (online mode)

You can also test a gateway configuration online by establishing a link between MSI.designer and a MSI 400 system.

Read here how to activate the online mode and what you need to take into account: Software manual, chapter "Connecting to the safety controller"

**NOTICE**

**LED behavior for active connections**

If you are linked to a MSI 400 installation, the status LEDs in the **Module** view of MSI.designer will light up in the same way as for the connected system.

Further information about the status LEDs may be found in the documentation for the relevant module:

- *Modbus TCP gateway [chapter 6.4]*
- *PROFIBUS-DB gateway [chapter 9.4]*
- *CANopen gateway [chapter 10.15]*

# 6 Modbus TCP gateway

The controller module MSI 430-x/???- LEUZE can be used for Modbus TCP. The internal MSI-EN-MOD module (Modbus TCP Gateway) is a component of the MSI 430-x/???- LEUZE device and is activated by the gateway configuration:

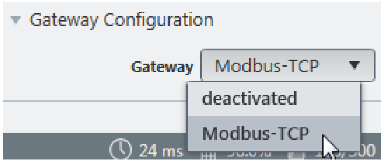



Fig. 6.1: Activation of the Modbus TCP on the MSI 430-x and ???- LEUZE modules

The Modbus TCP gateway supports the following:

- Modbus TCP with master and slave operation
- Ethernet TCP/IP socket interface, polling and auto-update function

## 6.1 Interfaces and operation

MSI 430-x and ???- LEUZE are equipped with an RJ-45 socket.

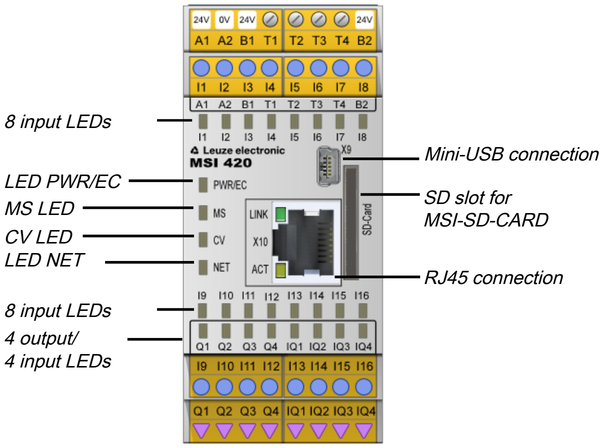



Fig. 6.2: Interfaces and display elements

### Further information

- Here in this manual:
  *Diagnostics and troubleshooting [chapter 6.4]*
- In the hardware manual:
  Device state and LED displays in the controller modules

## 6.2 Basic configuration – allocation of an IP address

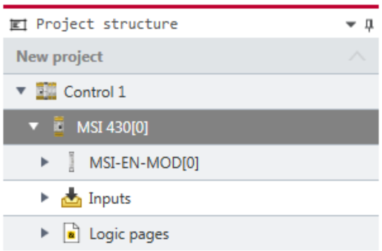
The MSI 430-x/???- LEUZE module is configured with the aid of the MSI.designer configuration software.

### Step 1: Insert module MSI 430

- ↳ Start MSI.designer and open the **Modules** docking window.
- ↳ Add the controller module MSI 430-x/???- LEUZE.
  Instructions: Software manual, chapter "Adding modules"

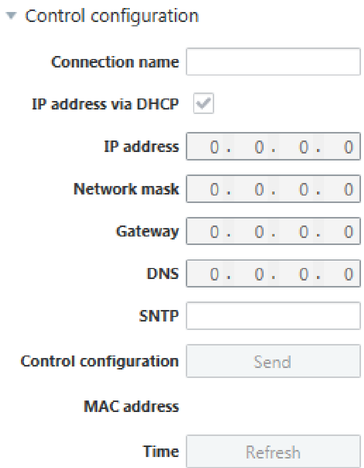
### Step 2: Open configuration dialog

- ↳ Switch to the **Project structure** docking window.
- ↳ Click on the top element, which represents the controller.




↳ Open the **Properties** docking window.

⇨ You will see the controller configuration dialog.




### Step 3: Store configuration

↳ Enter the following values under **Controller configuration**:
- valid IP address
- subnet mask
- if required: valid IP address for a default gateway
OR:
Alternatively activate DHCP.

↳ Ensure that MSI.designer is connected to the MSI 400 system.
The MSI 400 system must not be in **Run (Execute)** mode. The **Start/Stop** button in the command bar of the **Modules** view must be set to **Start**.


More detailed information on connecting with the controller: Software manual, chapter "Connecting to the safety controller"

↳ Click **Send** in the **Properties** docking window to transfer the configuration to the MSI 400 system.

## 6.3 Configuration of the Modbus-TCP interface to the PLC - how the data are transferred

**Application characteristics for Modbus TCP**

- Support of standard addressing conventions for Modbus TCP
- Master and slave operation

**Requirements for the PLC for Modbus TCP**

- The PLC must support the Modbus TCP protocol.
- The PLC must either support the Read Holding Registers and Write Multiple Registers commands or the Read/Write Multiple Registers command.

The configuration steps in this section determine how the data are to be transmitted to the higher-level PLC.

There are two different methods of transmission for each transmission direction, i.e. MSI 400 **to network** and **network to** MSI 400:

- Receiving method Polling/PLC requests (gateway as slave)

  This method allows the PLC regularly to request data using polling.

  When this method is used, the data are returned in the response to the data request. The PLC requests data by accessing the receiving data address of the MSI 430-x/???- LEUZE module by means of a read holding register telegram.
- The master receiving method gateway writes to the PLC (auto-update, gateway as master)

  When the MSI-EN-MOD module sends data to the PLC, these are immediately written to a memory location in the PLC.
- Slave transmission method - PLC writes (gateway as slave)

  With this method, the PLC sends telegrams to the MSI 430-x/???- LEUZE module to write to the output data sets. For this purpose, the PLC writes data into defined addresses.
- The master transmission method gateway reads from the PLC (auto-update, gateway as master)

  With the master transmission method, the MSI 430-x/???- LEUZE module polls the PLC for the output data sets.

**NOTICE**

The configuration is regarded as faulty when the IP address of the PLC is zero and the read transfer mode and/or write transfer mode has been set for the master.

The number of possible connections to the PLC depends on whether the MSI 430-x/???- LEUZE module is operated as a master or as a slave. Depending on the setting, up to 6 PLCs can simultaneously address the MSI 430-x/???- LEUZE module.

Tab. 6.1: Maximum number of possible Modbus TCP connections for the individual operating modes

| Operating mode of the MSI 430-x/???- LEUZE module | Maximum number of connections |
|---|---|
| Output data (to PLC): Gateway writes<br>Input data (from PLC): Gateway reads | 1 outgoing connection<br>1 incoming connection |
| Output data (to PLC): Gateway writes<br>Input data (from PLC): PLC writes | 1 outgoing connection<br>6 incoming connections |
| Output data (to PLC): PLC reads<br>Input data (from PLC): Gateway reads | 6 outgoing connections<br>1 incoming connection |
| Output data (to PLC): PLC reads<br>Input data (from PLC): PLC writes | 6 outgoing connections<br>6 incoming connections |

The following table describes the configuration, depending on the transmission method:

### Gateway is master

Tab. 6.2: Configuration directive – gateway as master

| Essential settings in the gateway configuration (via MSI 430-x/???- LEUZE) | Settings required for the PLC program and/or in the Modbus TCP configuration tool |
|---|---|
| Choose **Gateway writes to tag/file** and/or **Gateway reads from register** to configure the gateway as a master. | – |
| Select which data are to be written to the PLC or read from it. | – |
| Define where the selected data in the PLC memory are to be written to: Enter the register address(es).<br>Example: "40001" and/or you can determine from which location in the PLC memory the selected data are to be read: Enter the register addresses. | Ensure that the addresses allocated in the MSI 400 are available and that they contain the data intended for the MSI 400- system. |
| Choose how often these data are to be transmitted. | – |
| Define from and to where the data in the Modbus-TCP network are to be read and written: Enter the IP address and the slot number of the PLC controller. | – |

### Gateway as slave

Tab. 6.3: Configuration directive – gateway as slave

| Essential settings in the gateway configuration (via MSI 430-x/???- LEUZE) | Settings required for the PLC program and/or in the Modbus TCP configuration tool |
|---|---|
| Select **PLC requests** and **PLC writes** in the gateway configuration dialog. | – |
| – | Select which data are to be written to the gateway or read from it.<br>Ensure that the PLC program writes the data into the addresses allocated to the gateway (see Table "*Data addressing for the MSI 430-x/???- LEUZE as receiver [chapter 6.3]*"). |

**NOTICE**

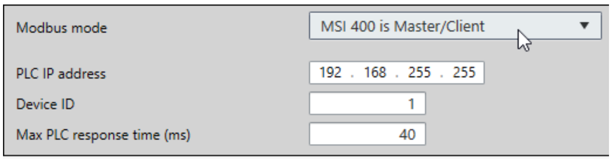
The address settings for the Modbus TCP gateway are 1-based. Please subtract 1 from the register address set in MSI.designer for a 0-based address setting.

Example: Register 1100 corresponds to the Modbus address 1099.

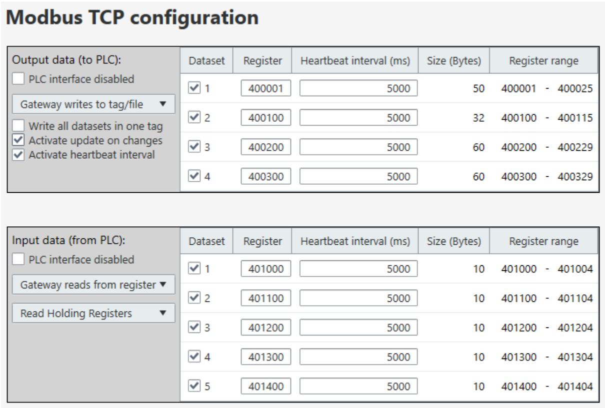
### Master mode: MSI 430 reads from/writes to the PLC

Carry out the following steps to configure the gateway as a master:

- Change to the **Gateway** view and click on the **Gateway configuration** tab.
- Select the MSI 400 is Master/Client option at the very bottom of the **Modbus mode** section.




⇨ The settings required for master mode are activated automatically in the **Output data (to PLC)** and **Input data (from PLC)** sections.




### Quick reference

You can make the following additional settings:

| **“Output data (to PLC)” section** | |
|---|---|
| **Area highlighted in gray** | |
| Selection list | Set automatically: Determines the transmission method.<br>Value required for master mode:<br>Gateway writes to tag/file |
| **All data sets in one tag** | Optional<br>Defines that all data sets are to be written to a single address in the PLC memory.<br>In this case the register address defined for Data Set 1 will be used. |
| | **Note:**<br>The following two settings can be activated simultaneously. They determine the frequency of data transmission. |
| **Activating updates following changes** | Recommended<br>Determines that the MSI 430-x/???- LEUZE module immediately updates the data in the PLC as soon as changes are made to the data sets. |
| **Activate heartbeat interval** | Recommended<br>Use the heartbeat intervals which you defined in the **Heartbeat interval** column to activate the update of the selected data sets. |
| **Columns highlighted white** | |
| **Data set** | Determines which data are to be written to the PLC or read from it.<br>Select the checkboxes for the desired data sets.<br>You will find a detailed description of the data sets here: *Data transferred to the network (network input data sets) [chapter 3.2]* |
| **Register** | Define from and to where in the PLC memory the selected data should be read and written. |
| **Heartbeat interval (ms)** | Defines how often the data sets are to be updated.<br>Requirement: You have selected the option **Activate heartbeat interval** (see above). |
| **Register range** | Shows the registers in the PLC to which the process data is writ-ten. |
| “Input data (from PLC)” section | |

| “Output data (to PLC)” section | |
|---|---|
| Selection list 1 | Set automatically: Determines the transmission method.<br>Value required for the master mode:<br>Gateway reads from register |
| Selection list 2 | Defines which of the two modbus commands is used:<br>• **Read holding registers**:<br>Activates the **Read holding registers** command (see "Module commands" table below).<br>• **Read input registers**:<br>Activates the **Read input registers** command (see "Module commands" table below). |
| **Data set** column | Determines which data are to be written to the PLC or read from it.<br>Mark the control boxes for the desired data sets for this purpose.<br>You will find a detailed description of the data sets here: *Data transferred to the network (network input data sets) [chapter 3.2]* |
| **Register** column | Define from and to where in the PLC memory the selected data should be read and written. |
| Column **heartbeat interval** | Defines how often the data sets are to be updated. |
| “Modbus mode” section | |
| **PLC IP address**<br>**Controller ID** | The parameters define from and to where the data in the Modbus-TCP network are to be read and written: |
| **Maximum refresh time for PLC** | Define the maximum rate (or the minimum time interval) for transmitting the data sets to the PLC. This setting depends on the processing speed of the PLC. Minimum = 10 ms, maximum = 65535 ms. The basic setting of 40 ms is suitable for most PLC<br>Note: When these values are greater than the heartbeat interval, the **heartbeat interval** will be slowed down to this value. |

↳ Connect MSI.designer with the MSI 400 system and transmit the configuration.
More information on connecting with the controller: Software manual, chapter "Connecting to the safety controller"

**Write to the PLC**

| | ***NOTICE*** |
|---|---|
| ! | The following restrictions apply when the gateway operates as a master and writes the input data sets to the PLC: |

- The address of the input data sets (preset in MSI.designer) must be the same as defined in the PLC.
- The PLC variable that is to incorporate the data must meet the following conditions:
  - in the address range 40xxxx (for Schneider Modicon PLC),
  - an array of 16-bit words,
  - long enough to contain the defined input data set array.
- All input data sets are transmitted to the PLC in 16-bit word format, with the first byte having the lowest value, i.e. on the far right of the integer, while the second byte has the highest value, i.e. on the very left of the integer.

### Reading from the PLC

**NOTICE**

The following restrictions apply when the gateway operates as a master and reads the output data sets from the PLC:

- The address of the output data sets must be the same as defined in the PLC.
  Please note: The value of the Modbus addresses must be 1 lower than the register data. See also: "Figure 8" in "MODBUS Application Protocol V1.1b3"
- The PLC variable from which the data are requested must meet the following conditions:
  - They fall into the address range 40xxxx (for Schneider Modicon PLCs).
  - There is an array of 16-bit words for the output data sets that is long enough to accommodate the entire output data set.
- All output data sets are transmitted to the PLC in 16-bit word format, with the first byte having to be placed as the lowest value, i.e. on the far right of the integer, while the second byte will have the highest value, i.e. on the very left of the integer.

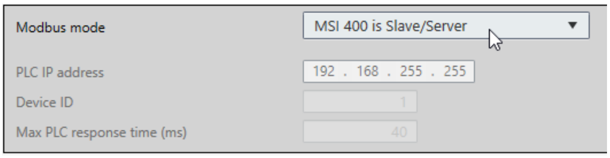
### Slave/server mode - PLC reads from / writes to MSI 430

In this operating mode, the MSI 430-x/???- LEUZE module provides the data as a slave at the request of the PLC. If this operating mode is desired:

↳ Launch MSI.designer.

↳ Change to the **Gateway** view and click on the **Gateway configuration** tab.

↳ Select the MSI 400 is Slave/Server option at the very bottom of the **Modbus mode** section.




⇨ The minimum settings required for slave mode are activated automatically in the **Output data (to PLC)** and **Input data (from PLC)** sections.

⇨ Unavailable options are grayed out.

You can make the following additional settings:

Tab. 6.4: “Output data (to PLC)” and “Input data (from PLC)” sections

| Setting | Description/procedure |
|---|---|
| **Data set** column | Determines which data can be written to the PLC or read from it.<br>Mark the control boxes for the desired data sets for this purpose.<br>You will find a detailed description of the data sets here: *Data transferred to the network (network input data sets) [chapter 3.2]* |
| **Size (bytes)** column | Exact number of bytes to be read out or written. The number of 16-bit data types usual for TCP modbus is exactly half. |
| **Register range** column | Registers to be addressed in MSI 430-x/???- LEUZE |

↳ Connect MSI.designer with the MSI 400 system and transmit the configuration.
More information on connecting with the controller: Software manual, chapter "Connecting to the safety controller"

### PLC writes output data sets

The following restrictions apply when the PLC writes the output data sets:

- The equipment index must not be equal to zero.
- The telegram must be sent in Word format.
- All output data sets are transmitted to the PLC in 16-bit word format, with the first byte having to be placed as the lowest value, i.e. on the far right of the integer, while the second byte will have the highest value, i.e. on the very left of the integer.

### PLC polls the input data sets

- The following restrictions apply:
- The equipment index must not be equal to zero.
- The PLC variable that is to incorporate the data must meet the following conditions:
  - It falls into the address range 40xxxx (for Schneider Modicon PLCs).
  - There is an array of 16-bit words that is long enough to accommodate the entire output data set.
- All input data sets are transmitted to the PLC in 16-bit word format, with the first byte having the lowest value, i.e. on the far right of the integer, while the second byte has the highest value, i.e. on the very left of the integer.

**NOTICE**

Configure the PLC data polling in such a way that a data telegram is exchanged at least once a minute between MSI 430-x/???- LEUZE and the PLC. The TCP connection will otherwise be interpreted as not used and terminated.

**NOTICE**

The data from the PLC to the MSI 430-x/???- LEUZE module assumes the value zero in the MSI.designer logic program if the Modbus TCP connection is terminated by the PLC itself or by a timeout.

### MSI 430 as slave – data addressing

The following table lists the addresses for reading out the data sets.

### Unit ID 1

Tab. 6.5: Data addressing for the MSI 430-x/???- LEUZE as receiver

| Register (Base 1) | Description | Access | Scope (words) |
|---|---|---|---|
| 1000 | Request data for all activated input data sets | Read | 1..101 [1)] |
| 1100 | Request data from input data block 1-5 | Read | 1..25 |
| 1200 | Request CRC data | Read | 1..16 |
| 1300 | Request diagnostic data | Read | 1..30 |
| 1400 | Reserved | Read | 1..30 |
| 2000 | Write all activated output data sets | Read, write | 1..25 [2)] |
| 2100 | Write data from output data set 1 | Read, write | 1..5 |
| 2200 | Write data from output data set 2 | Read, write | 1..5 |
| 2300 | Write data from output data set 3 | Read, write | 1..5 |
| 2400 | Write data from output data set 4 | Read, write | 1..5 |
| 2500 | Write data from output data set 5 | Read, write | 1..5 |

[1)] Corresponds to all activated input data sets.

[2)] Must correspond to all activated output data sets. Example: If only output data sets 1 and 2 have been activated, 10 words (20 bytes) must be written. If all output data sets have been activated, 25 words (50 bytes) must be written.

### Modbus commands and error messages

The MSI 430-x/???- LEUZE module supports the following Modbus commands and error messages:

Tab. 6.6: Modbus commands

| Modbus command | Value |
|---|---|
| Read holding registers | 3 |
| Read input[1)] registers | 4 |
| Write single register | 6 |
| Write multiple registers | 16 (10hex) |
| Read/write multiple registers | 23 (17hex) |
| [1)] starting with module version A-03 | |

Tab. 6.7: Modbus error messages

| Modbus error response | Description |
|---|---|
| 1 Function not permitted | The requested function is not supported |
| 2 Data address not permitted | Undefined data address received |
| 3 Data value not permitted | Request with prohibited data values, e.g. insufficient data requested for a data set |
| 4 server errors | An error occurred during execution of the server. |

## 6.4 Diagnostics and troubleshooting

You can find information about the diagnostics of the MSI 400 system in the software manual.

Tab. 6.8: Troubleshooting on MSI 430-x/???- LEUZE

| Error | Possible cause | Possible remedy |
|---|---|---|
| Key: ○ LED off / ☀ LED flashes / ● LED lights up | | |
| MSI.designer cannot set up a connection to the MSI 400 gateway. | • The MSI 430-x/???-LEUZE module has no power supply.<br>• The MSI 430-x/???-LEUZE module is not in the same physical network as the PC.<br>• A different subnet mask has been set in the TCP/IP settings for the PC.<br>• The module was been preconfigured and has a permanently set IP address or an IP address allocated to a DHCP server that has not been allocated. | • Switch on the power supply. Check the Ethernet wiring and the network settings of the P and correct them where necessary.<br>• Set the PC to a network address 192.168.1.0 (For module MSI 420/430, the delivery state of the SD card sets address 192.168.1.5, which may not be used for the PC.)<br>• Alternatively activate DHCP on the PC and link the MSI 430-x/???-LEUZE module and the PC to a network, using an active DHCP server. (The delivery state of the SD card activates a DHCP client on the MSI 420/430 module. If no DHCP server is found within about 1 minute during an active network connection, the address 192.168.1.5 is set in the MSI 420/430 module.)<br>• Check the communication settings in MSI.designer. |
| The MSI 430-x/???- LEUZE module does not provide any data.<br>LED PWR/EC: ● Green<br>LED LINK: ● Green<br>LED ACT: ☀ Yellow | • The MSI 430-x/???-LEUZE module has been configured for data transmission to the PLC, but no Ethernet communication has been established or it is faulty.<br>• Duplicate IP address detected. Another network device has the same IP address. | At least one Ethernet link must be established.<br>• Set up the Ethernet link on the PC, check the Ethernet wiring, check the Ethernet settings in the PLC and in MSI.designer.<br>• If no Ethernet communication is required, deactivate the Ethernet connections / PLC interfaces on the MSI 430-x/???- LEUZE module.<br>• Correct the IP address and switch the system off and on again. |

| Error | Possible cause | Possible remedy |
|---|---|---|
| The MSI 430-x/???- LEUZE module does not provide any data.<br>LED PWR/EC: Green<br>LED LINK: Green<br>LED ACT: Yellow<br>MS LED: Red (1 Hz) | • Configuration required.<br>• The configuration has not yet been fully transmitted.<br>• The module version of the controller module does not support the gateway function. | • Configure the MSI 430-x/???- LEUZE module and transfer the configuration to the device.<br>• Wait until the configuration has been fully transferred.<br>• Use the controller module with the required module version. |
| The MSI 430-x/???- LEUZE module does not provide any data.<br>LED PWR/EC: Green<br>LED LINK: Green<br>LED ACT: Yellow<br>MS LED: Green | • No data set was activated.<br>• No Ethernet communication interface was activated. | • Activate at least one data set. |
| The MSI 430-x/???- LEUZE module does not provide any data.<br>LED PWR/EC: Green<br>MS LED: Green (1 Hz) | The MSI 430-x/???- LEUZE module is in the "Stop" state. | The controller module is stopped.<br>• Start the controller module (switch to Run mode). |
| The MSI 430-x module functioned correctly after configuration but suddenly provides no more data.<br>LED PWR/EC: Green<br>LED LINK: Green<br>LED ACT: Yellow<br>MS LED: Green | • The MSI 430-x/???- LEUZE module is operated in slave mode, the IP address is allocated by a DHCP server.<br>• Following a restart of the MSI 430-x/???- LEUZE module or the DHCP server, another address was allocated to the MSI 430-x/???- LEUZE module, which is unknown to the PLC. | • Allocate a fixed IP address to the MSI 430-x/???- LEUZE module.<br>**or**<br>• Reserve a fixed IP address for the MSI 430-x/???- LEUZE module in the DHCP server (manual assignment using the MAC address of the MSI 430-x/???- LEUZE module). |

<table>
<tr><th colspan="2">Error</th><th>Possible cause</th><th>Possible remedy</th></tr>
<tr><td colspan="2">The MSI 430-x/???- LEUZE module / the MSI 400 system is in the “Critical error” status</td><td rowspan="5"><ul><li>The MSI 430-x/???- LEUZE module is not properly connected to the other modules.</li><li>The module connection plug is dirty or damaged.</li><li>Another module in the MSI 400 system has an internal critical error.</li><li>The voltage supply for the MSI 430-x/???- LEUZE module is or was outside the specifications.</li></ul></td><td rowspan="5"><ul><li>Plug in the MSI 430-x/???- LEUZE module correctly.</li><li>Clean the connection plug and socket.</li><li>Switch on the power supply again.</li><li>Check the power supply.</li><li>Check the other modules of the MSI 400 system.</li></ul></td></tr>
<tr><td>LED PWR/ EC</td><td>Red</td></tr>
<tr><td>LED LINK</td><td>Green</td></tr>
<tr><td>LED ACT</td><td>Yellow</td></tr>
<tr><td>MS LED</td><td>Red</td></tr>
</table>

## 6.5 Status bits

The Modbus TCP Gateway MSI-EN-MOD sets status bits, which are available in the logic editor of MSI.designer for processing.

Tab. 6.9: Meaning of the status bits MSI-EN-MOD[0] in the logic editor

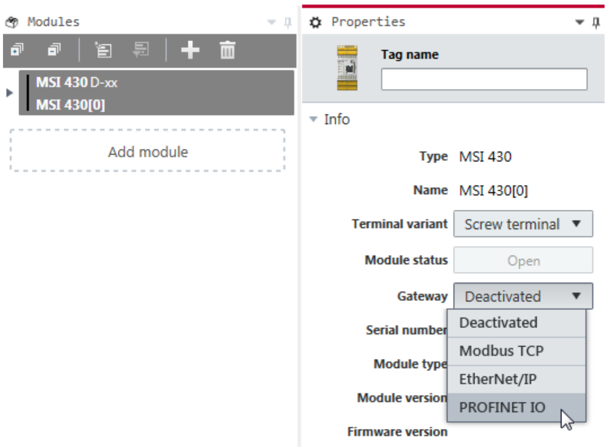
| Name of the status bits | Set to 1, if ... | Reset to 0 ... |
|---|---|---|
| Output status | ... At least one output data byte was sent without error. | ... If there is a missing Modbus TCP connection to the PLC. |
| Input status | ... At least one input data byte was sent without error. | ... If there is a missing Modbus TCP connection to the PLC. |
| Internal state | … the Modbus function in MSI 430-x/???- LEUZE is ready for communication<br>From module version E-01.01:<br>if the Modbus function in MSI 430-x/???- LEUZE is ready for communication or if at least one input or output byte has been downloaded or transmitted without errors. | ... If an error has occurred in the Modbus function. |

# 7 PROFINET IO-Gateway

The MSI 430-x/???- LEUZE module can be used for PROFINET IO.

You will find the GSDM file and the equipment symbol for integration into a PLC of the product website of the MSI 430-x/???- LEUZE module on the Internet (www.leuze.com).

The internal MSI-EN-PN (PROFINET IO Gateway) module is part of the MSI 430-x/???- LEUZE device. You can activate it in the configuration dialog of the MSI 430-x/???- LEUZE module in the **Properties** docking window:




### Supported features

The MSI 430-x/???- LEUZE module supports:

- PROFINET IO Conformance Class A
- Cyclical IO communication (RT)
- LLDP
- DCP
- Auto MDI
- Auto negotiation
- I&M 1-4
- Equipment diagnostics, alarms

Currently not supported:

- SNMP
- Shared Input, Shared Device
- FSU
- MIB II
- Port statistics

The number of PROFINET controllers (PLCs) which can simultaneously connect to a MSI 430-x/???- LEUZE device via PROFINET is limited to one.

## 7.1 Interfaces and operation

Interfaces and operation are identical to that of the Modbus TCP Gateway.

Read the following section: *Interfaces and operation [chapter 6.1]*

## 7.2 Basic configuration - Assigning a device name and an IP address

Configuration and diagnostics of the MSI 430-x/???- LEUZE module is possible both with the help of the MSI.designer configuration software and with the PROFINET IO network programming tool (e.g. SIEMENS TIA Portal).

### Configuration using PROFINET IO

In the delivery state, a MAC address is stored in every PROFINET IO field device such as the MSI 430-x/???- LEUZE module. The symbolic name (NameOfStation) **Test station** is stored on the SD card in the delivery state.

> **NOTICE**
>
> - In accordance with IEC 61158-6-10 no capital letters are permitted for the symbolic name (NameOfStation).
> - This NameOfStation is used by the I/O controller (e.g. the PLC) to assign an IP address to the field device.
> - If the IP address is also used for other communication via Ethernet, such as TCP/IP or for the configuration via Ethernet, please note that the PLC changes the IP address and can thus interrupt the other communication.

The IP address is assigned in two steps.

- Assign a unique system-specific name to the Gateway, using either the network configuration tool such as SIEMENS TIA Portal, or using the MSI.designer software. In MSI.designer this is the **Connection name**.
  Where do you edit the connection name in MSI.designer? Open the **Project structure** docking window and there click on the **Controller** entry right at the top. Additionally open the **Properties** docking window and enter the desired value there in the configuration dialog under **Connection name**.
- A (unique) system-specific name can be used by the I/O-Controller (i.e. the PLC) to assign the IP address to the gateway now before the system is booted.

> **NOTICE**
>
> The MAC address of the MSI 430-x/???- LEUZE module is printed on the device's nameplate (example: 00:07:17:02:03:05).

### Using the Siemens TIA Portal to assign device names

In the **Online accesses** area, select the network card connected to the network which can be used to access the MSI 430-x/???- LEUZE device. In the **Assign name** function area, edit the **PROFINET device name** field and then select **Assign name**.

This will permanently assign the new device name to the MSI 430-x/???- LEUZE device.

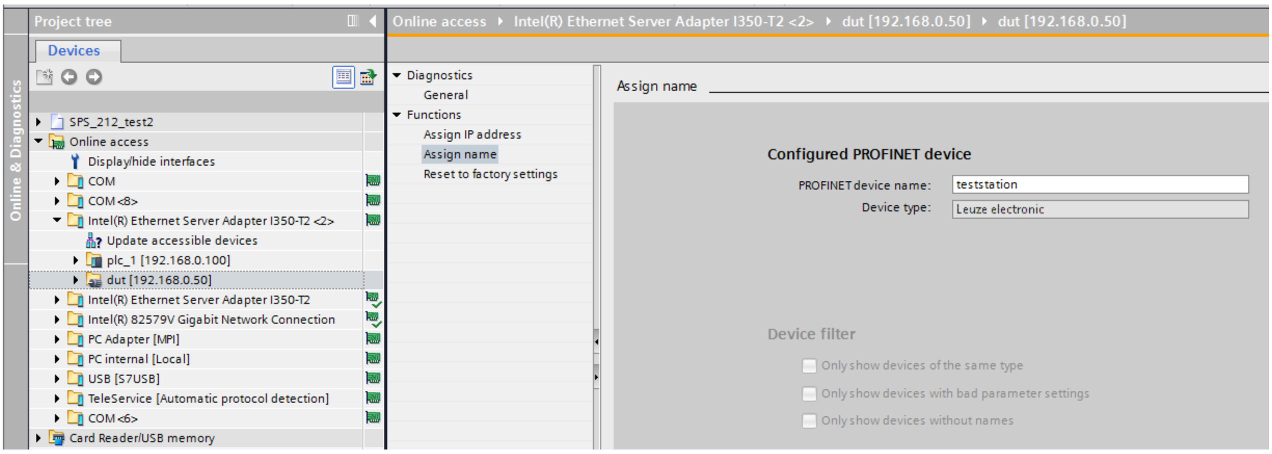



Fig. 7.1: Assigning device names with the TIA portal

### Assign device name via the software

- Launch MSI.designer and connect to the controller module MSI 430-x/???- LEUZE.
- Press the **Stop** button in the **Modules** view to stop the application.

- In the **Modules** view click on the blue background and open the **Properties** docking window.
  - ⇨ You will see the controller configuration dialog.
- Edit the connection name and click the **Send** button.

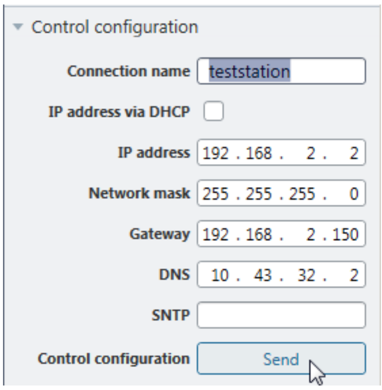



Fig. 7.2: Configuration dialog for the IP data and the device name

**NOTICE**

- The format of the device name must correspond to the specification of the PROFINET standard.
- Ensure that the address for the default gateway matches the address set by the PLC for the gateway. If no router is used, then Siemens Step 7 uses the same IP address for the default gateway as for the MSI 430-x/???- LEUZE module.
- If a project file with an active PROFINET IO is provided on the MSI 430-x/???- LEUZE module, only one device in MSI.designer can be found by USB. If you would like to use the Ethernet to connect to the MSI 430-x/???- LEUZE module, select **Edit** in the **Connect** dialog, where you then set the IP address of the MSI 430-x/???- LEUZE module.

#### Set the IP address using the software

The IP address is typically assigned by the PROFINET IO controller (e.g. PLC). However, the MSI 430-x/???- LEUZE module also allows configuration of the entire MSI 400 system via Ethernet TCP/IP. In this case, it may be necessary to already assign an IP address to the MSI 430-x/???- LEUZE module before the PROFINET IO network is set up. This can also be done in the configuration dialog shown above.

## 7.3 PROFINET configuration of the gateway - how the data are transferred

The following steps are required to configure communication between the PLC and the gateway.

**NOTICE**

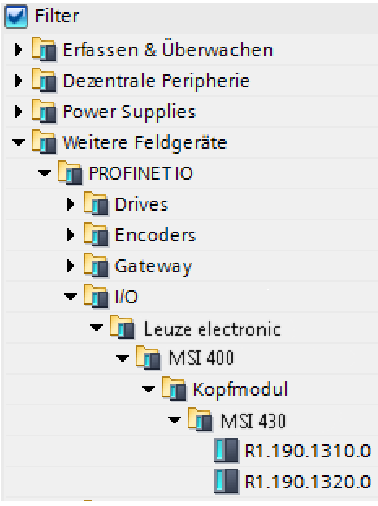
This documentation does not address the installation of the PROFINET IO network or the other components of the automation system project in the network configuration tool. It is assumed that the PROFINET project in the configuration program, e.g. the SIEMENS TIA Portal, has already been set up. The examples presented are based on configurations created with the help of the SIEMENS TIA Portal.

#### Step 1: Install the device master file (GDSML file)

Before the MSI 430-x/???- LEUZE module can be used for the first time as part of the network configuration tool, e.g. the SIEMENS TIA Portal, the gateway's device master file (GSDML file) must first be installed in the tool's hardware catalog.

- Download the GSDML file and the equipment symbol of the MSI 430-x/???- LEUZE module from the product site (www.leuze-shop.com).
- Follow the instructions for installing GSDs in the online help section or in the user manual for the PROFINET network configuration tool.

If you are using SIEMENS TIA Portal, the MSI 430-x/???- LEUZE module will appear in the following location in the hardware catalog:



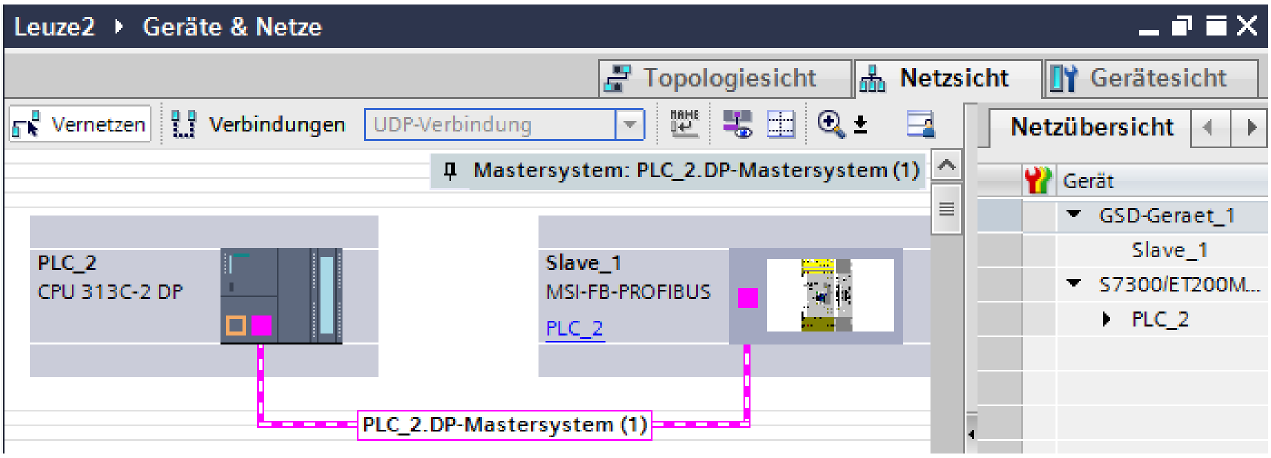

**Step 2: Add the gateway to the project**

To make the system of the MSI 400 system available in the process diagram of the PLC, the gateway must first be added to the hardware configuration. The procedure to be used depends on the hardware configuration software of the PLC used. Please also read the documentation for the corresponding software in this regard.

The example below shows how the gateway is added to a SIEMENS TIA Portal project.

↳ Use Drag & Drop in the **Network view** to drag the device to the Ethernet PROFINET IO network.

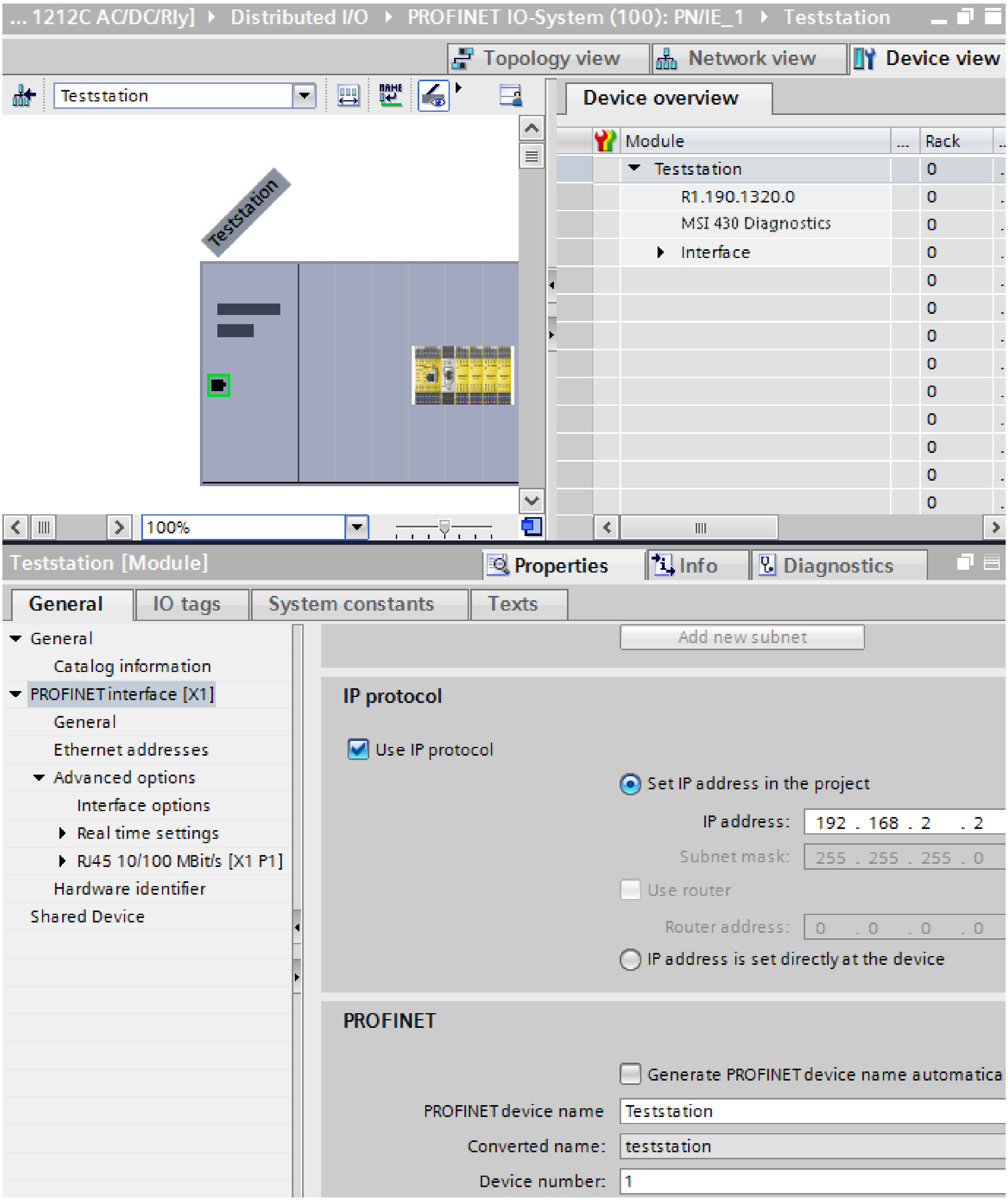
Example:




**Step 3: Configure the gateway properties**

↳ Double-click the hardware symbol of the MSI 430-x/???- LEUZE module.

↳ Configure the IP address, the device name, and the update interval of the cyclical I/O data exchange. Select the **Properties** tab for this.




**NOTICE**

The PLC can only communicate with the MSI 430-x/???- LEUZE module if the PLC software and the gateway use the same gateway name.
In accordance with IEC 61158-6-10 no capital letters are permitted for the symbolic name (NameOfStation).

## 7.4 PROFINET configuration of the gateway - which data are transferred

### Cyclical data

The physical I/O modules are not presented in the PROFINET IO hardware catalog as typical hardware modules in the network. Instead, the data provided by the MSI 400 system has been arranged in various data blocks. Every data block represents a module in the PROFINET IO hardware catalog. The GSDML supports 13 Slots in which the modules can be placed. This makes is possible to use each data set one time (see illustration *"Configuration" [chapter 7.4]*).

### Process data from module to PLC

The MSI 430-x/???- LEUZE module provides 5 input data blocks (virtual device modules) which contain the process image. These can be exclusively placed in each corresponding slot 16 to 20.

***NOTICE***

Input data blocks 1 to 4 each contain 12 bytes, while input data block 5 contains 2 bytes.

The content of the input data blocks can be freely selected. The data assignment in MSI.designer is pre-configured in accordance with the following:

Tab. 7.1: Predefined content of input data block 1 to 5 of the MSI 430-x/???- LEUZE module

| | Data block 1 | Data block 2 | Data block 3 | Data block 4 | Data block 5 |
|---|---|---|---|---|---|
| Byte no. per data block | Input data | Input data | Input data | Input data | Input data |
| Byte 0 | MSI 430-x/???- LEUZE input values | I/O module 1 input values | I/O module 1 output values | Not allocated | Not allocated |
| Byte 1 | MSI 430-x/???- LEUZE input values | I/O module 2 input values | I/O module 2 output values | Not allocated | Not allocated |
| Byte 2 | MSI 430-x/???- LEUZE input values | I/O module 3 input values | I/O module 3 output values | Not allocated | Not available |
| Byte 3 | MSI 430-x/???- LEUZE output values | I/O module 4 input values | I/O module 4 output values | Not allocated | |
| Byte 4 | Logic data values | I/O module 5 input values | I/O module 5 output values | Not allocated | |
| Byte 5 | Logic data values | I/O module 6 input values | I/O module 6 output values | Not allocated | |
| Byte 6 | Logic data values | I/O module 7 input values | I/O module 7 output values | Not allocated | |
| Byte 7 | Logic data values | I/O module 8 input values | I/O module 8 output values | Not allocated | |
| Byte 8 | Logic data values | I/O module 9 input values | I/O module 9 output values | Not allocated | |
| Byte 9 | Logic data values | I/O module 10 input values | I/O module 10 output values | Not allocated | |
| Byte 10 | Logic data values | I/O module 11 input values | I/O module 11 output values | Not allocated | |
| Byte 11 | Logic data values | I/O module 12 input values | I/O module 12 output values | Not allocated | |
| Length | 12 bytes | 12 bytes | 12 bytes | 12 bytes | 2 bytes |

| | Data block 1 | Data block 2 | Data block 3 | Data block 4 | Data block 5 |
|---|---|---|---|---|---|
| Byte offset | 0 | 12 | 24 | 36 | 48 |

1 byte for data set 1 is available for every expansion module. The input values show the state of the preliminary evaluation of the I/O module. This corresponds to the state of the element in the controller module logic. The level at the associated terminal cannot be clearly detected from this, as the data may be set to low, irrespectively of the level at the input terminal, by means of the cross-connection detection or two-channel evaluation (e.g. I1-18).

When two-channel input elements have been configured for an I/O module, only the lower-value bit represents the pre-evaluation state of the corresponding element (e.g. bit 0 for I1 and I2, bit 2 for I3 and I4, bit 4 for I5 and I6, bit 6 for I7 and I8).
The higher-value bit (bit 1, 3, 5 and 7) is used as follows in this case:
0 = error, 1 = no error

Further information

You will find information about how to configure the process diagram in the description of the (*The graphical user interface [chapter 5.1]*) user interface.

**Data from the PLC to the MSI 430 module**

There are 5 output data blocks having 10 bytes each. These can be exclusively placed in each corresponding slot 21 to 25.

The content of these data blocks can be used as input in the MSI.designer logic editor or forwarded to another network by a second gateway. Every bit to be used must be assigned a tag name in order to provide the desired bits in the logic editor or for forwarding. Bits without tag names are not available.

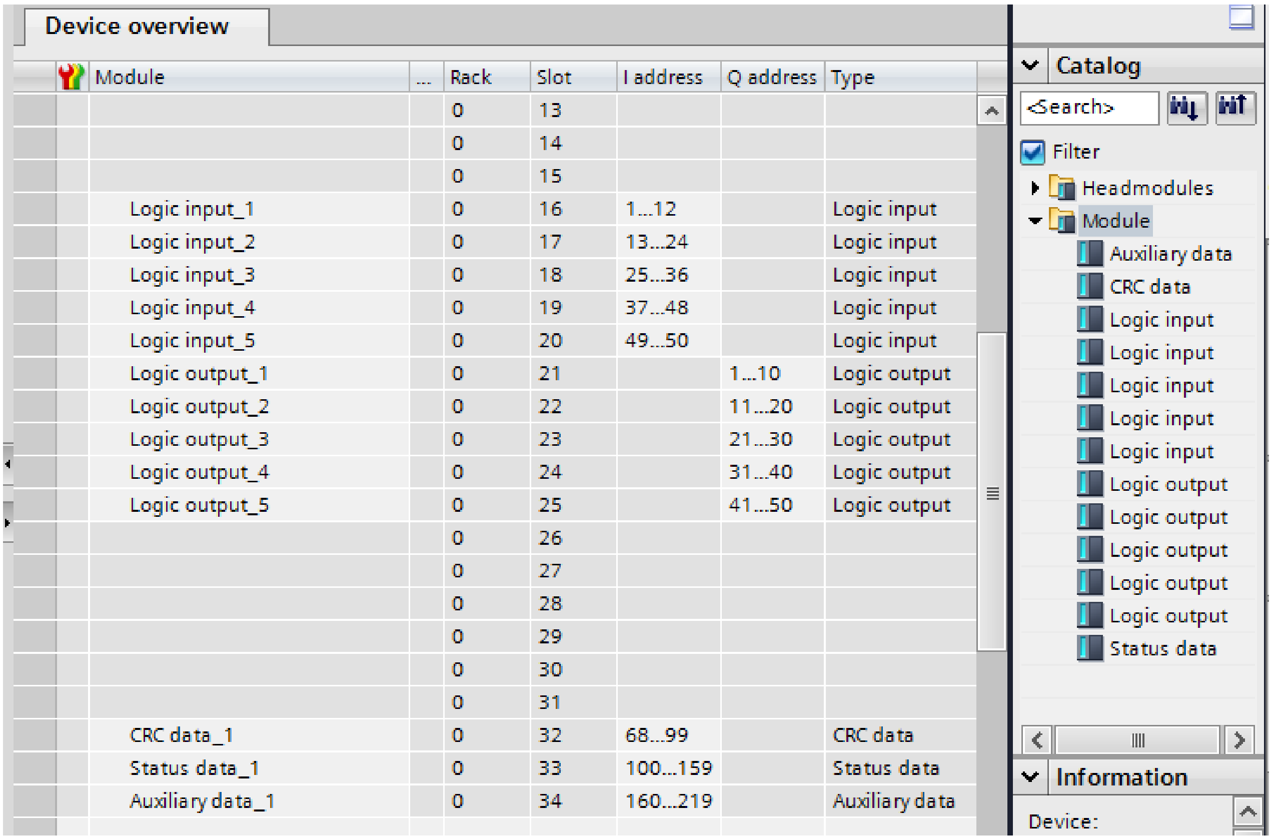
Detailed information about how you can assign and adapt the tag names of the input and output data can be found here:
Software manual, chapter "Adapting display names of project components"

> ***NOTICE***
>
> The standard value of the gateway data bit is zero following activation of the MSI 430-x/???-LEUZE device.
>
> If the connection to PLC is terminated, then all of the gateway data bits in the MSI.designer logic editor assume the value zero.

> ***NOTICE***
>
> For output data with IOPS=Bad, all of the gateway data bits in the MSI.designer logic editor assume the value zero. This is the case, for example, if the PLC is stopped.

**Settings in the PROFINET IO network configuration tool**

↳ Only drag the required data blocks from the hardware catalog of the SIEMENS TIA Portal to the corresponding slots of the MSI 430-x/???- LEUZE module within the configuration table.



Device overview

| Module | ... | Rack | Slot | I address | Q address | Type |
|---|---|---|---|---|---|---|
| | | 0 | 13 | | | |
| | | 0 | 14 | | | |
| | | 0 | 15 | | | |
| Logic input_1 | | 0 | 16 | 1...12 | | Logic input |
| Logic input_2 | | 0 | 17 | 13...24 | | Logic input |
| Logic input_3 | | 0 | 18 | 25...36 | | Logic input |
| Logic input_4 | | 0 | 19 | 37...48 | | Logic input |
| Logic input_5 | | 0 | 20 | 49...50 | | Logic input |
| Logic output_1 | | 0 | 21 | | 1...10 | Logic output |
| Logic output_2 | | 0 | 22 | | 11...20 | Logic output |
| Logic output_3 | | 0 | 23 | | 21...30 | Logic output |
| Logic output_4 | | 0 | 24 | | 31...40 | Logic output |
| Logic output_5 | | 0 | 25 | | 41...50 | Logic output |
| | | 0 | 26 | | | |
| | | 0 | 27 | | | |
| | | 0 | 28 | | | |
| | | 0 | 29 | | | |
| | | 0 | 30 | | | |
| | | 0 | 31 | | | |
| CRC data_1 | | 0 | 32 | 68...99 | | CRC data |
| Status data_1 | | 0 | 33 | 100...159 | | Status data |
| Auxiliary data_1 | | 0 | 34 | 160...219 | | Auxiliary data |



Fig. 7.3: Configuration of the MSI 430-x/???- LEUZE module

**NOTICE**

The input and output addresses indicate the location of the cyclical data in the memory. These can be addressed via the absolute addresses %I and %Q in the SIEMENS TIA portal.

### Acyclical data and alarms

#### Read out data

The PLC can read out the diagnostic data of the MSI 400 system. The diagnostic information is provided in three data sets, data sets 2, 3, and 4:

Data set 2 comprises 32 bytes and contains the project file's CRC 32. This can only be placed in slot 32.

Data set 3 comprises 60 bytes and contains the status of the MSI 430-x/???- LEUZE module and the individual I/O modules. This can only be placed in slot 33. See the following to interpret the status bits in data set 3: table *"Meaning of the module status bits of the controller module" [chapter 3.3.4]* and table *"Meaning of the module status bits of the IO modules" [chapter 3.3.4]*

Data set 4 (auxiliary data) comprises 60 bytes and is currently filled with reserved values. This can only be placed in slot 34.

**NOTICE**

Data set 4 in Slot 34 does not function with all versions of the SIEMENS TIA portal.

### Information & Management

The MSI 430-x/???- LEUZE module supports the I&M information defined in the PROFINET IO specification. The following I&M information can be read out:

Tab. 7.2: Readable I&M information

| Name | Size | Value range | I&M | Storage location |
|---|---|---|---|---|
| MANUFACTURER_ID (Vendor ID) | 2 bytes | 397 = 0x18D | 0 | MSI 430-x/???-LEUZE |
| ORDER_ID (Order ID) | 64 bytes | "50132988 + 51 blank spaces and "50132989 " + 51 blank spaces | 0 | MSI 430-x/???-LEUZE |
| SERIAL_NUMBER (IM_Serial_Number) | 8 bytes | "16010001" to "99129999" | 0 | MSI 430-x/???-LEUZE |
| HARDWARE_REVISION (IM_Hardware_Revision) | 2 bytes | 101 to 9999 | 0 | MSI 430-x/???-LEUZE |
| SOFTWARE_REVISION (IM_Software_Revision) | 6 to 9 Bytes | "V0.1.0" to "V99.99.99" | 0 | MSI 430-x/???-LEUZE |
| Device ID | | 05001 | 0 | MSI 430-x/???-LEUZE |
| REV_COUNTER (IM_Revision_Counter) | 2 bytes | 0 to 65535 | 0 | SD card |
| PROFILE_ID (IM_Profile_ID) | 2 bytes | 0x0000 (Non-profile) | 0 | MSI 430-x/???-LEUZE |
| PROFILE_SPECIFIC_TYPE (IM_Profile_Specific_Type) | 2 bytes | 0x0003 (IO modules) | 0 | MSI 430-x/???-LEUZE |
| IM_VERSION (IM_Version) | 2 bytes | 1 | 0 | MSI 430-x/???-LEUZE |
| IM_SUPPORTED (IM_Supported) | 2 bytes | 10 (= 0b1010) | 0 | MSI 430-x/???-LEUZE |
| TAG_FUNCTION | 32 bytes | 32 Bytes à 0x20..0x7E | 1 | SD card |
| TAG_LOCATION | 22 bytes | 32 Bytes à 0x20..0x7E | 1 | SD card |
| INSTALLATION_DATE (IM_Date) | 16 bytes | | 2 | SD card1) |
| DESCRIPTOR (IM_Descriptor) | 54 bytes | 54 Byte à 0x00..0xFF | 3 | SD card |
| IM_Signature | 54 bytes | 54 Byte à 0x00..0xFF | 4 | SD card |

1) Subject to changes

**Alarms**

Alarms can be acyclically read using the PROFINET IO alarm infrastructure. When an error in the MSI 400 system occurs, the PROFINET IO gateway sends a corresponding diagnostics alarm to the network. The details of the diagnostics alarm (text and help) are then available through the SIMATIC PLC interface. The RALRM (SFB54) function block in OB82 (diagnostics interrupt) allows you to make the details of the sent alarm directly available in the PLC program.

***NOTICE***

All alarms are output to module 0.

The cause of the alarm is displayed by an error message from the GSDML file.

The possible causes of an alarm can be found in the software manual, Section "List of all error messages".

## 7.5 Diagnostics and troubleshooting

Information about the diagnosis of the MSI 400 system can be found in the software manual, Section "List of all error messages".

Tab. 7.3: Troubleshooting on the MSI 430-x/???- LEUZE module

<table>
<tr><th colspan="2">Error</th><th>Possible cause</th><th>Possible remedy</th></tr>
<tr><td colspan="4">Key: ○ LED off / ✸ LED flashes / ● LED lights up</td></tr>
<tr><td colspan="2">The MSI 430-x/???- LEUZE module does not provide any data.</td><td rowspan="5"><ul><li>The MSI 430-x/???- LEUZE module has been configured for data transmission to the PLC, but no Ethernet communication has been established or it is faulty.</li><li>Duplicate IP address detected. Another network device has the same IP address.</li><li>Incorrectly formatted PROFINET device name</li></ul></td><td rowspan="5"><ul><li>PROFINET IO must be activated in the project file. At least one Ethernet link must be established. Check the Ethernet wiring, check the Ethernet settings in the PLC and in MSI.designer.</li><li>Correct the IP address and switch the system off and on again.</li><li>Compare the device name of the PROFINET master and the MSI 430-x/???- LEUZE module.</li></ul></td></tr>
<tr><td>LED PWR/ EC</td><td>● Green</td></tr>
<tr><td>LED LINK</td><td>● Green</td></tr>
<tr><td>LED /ACT</td><td>✸ Yellow</td></tr>
<tr><td>MS LED</td><td>● Green</td></tr>
<tr><td colspan="2">The MSI 430-x/???- LEUZE module does not provide any data.</td><td rowspan="5"><ul><li>Configuration required.</li><li>The configuration has not yet been fully transmitted.</li><li>The module version does not support any PROFINET IO.</li></ul></td><td rowspan="5"><ul><li>Configure the MSI 430- x/???- LEUZE module with a project file in which PROFINET IO is activated, and transfer the configuration to the MSI 430-x/???- LEUZE module.</li><li>Use a device starting with module MSI 430-x/???- LEUZE version B-xx.</li></ul></td></tr>
<tr><td>LED PWR/ EC</td><td>● Green</td></tr>
<tr><td>LED LINK</td><td>● Green</td></tr>
<tr><td>LED ACT</td><td>✸ Yellow</td></tr>
<tr><td>MS LED</td><td>✸ / ✸ Red/green</td></tr>
<tr><td colspan="2">The MSI 430-x/???- LEUZE module does not provide any data.</td><td rowspan="5"><ul><li>The MSI 400 system is in the stop state.</li></ul></td><td rowspan="5"><ul><li>Start the controller module (switch to Run mode).</li></ul></td></tr>
<tr><td>LED PWR</td><td>● Green</td></tr>
<tr><td>LED LINK</td><td>● Green</td></tr>
<tr><td>LED ACT</td><td>✸ Yellow</td></tr>
<tr><td>MS LED</td><td>✸ Green (1 Hz)</td></tr>
</table>

| Error | | Possible cause | Possible remedy |
|---|---|---|---|
| The MSI 430-x/???-LEUZE module does not provide any data. | | • The IP address for the MSI 430-x/???- LEUZE module is assigned by a DHCP server. Following a restart of the MSI 430-x/???- LEUZE module or the DHCP server, another address was allocated to the MSI 430-x/???- LEUZE module, which is unknown to the PLC. | • Either assign a permanent IP address to the MSI 430-x/???- LEUZE module or reserve a permanent IP address for the MSI 430-x/???- LEUZE module in the DHCP server (manual assignment using the MAC address of the MSI 430-x/???- LEUZE module). |
| LED PWR/ EC | Green | | |
| LED LINK | Green | | |
| LED ACT | Yellow | | |
| MS LED | Green | | |
| The MSI 430-x/???-LEUZE module / the MSI 400 system is in a Critical Error status. | | • The MSI 430-x/???-LEUZE module is not properly connected to the other MSI 400 modules.<br>• The module connection plug is dirty or damaged.<br>• Another MSI 400 module has an internal critical error. | • Insert the I/O module correctly. Clean the connection plug and socket.<br>• Switch on the power supply again.<br>• Check the other MSI 400 modules. |
| LED PWR | Green | | |
| LED LINK | Green | | |
| LED ACT | Yellow | | |
| MS LED | Red | | |

## 7.6 Deactivation of the PROFINET IO function

If the MSI 430-x/???- LEUZE device is started with an activated PROFINET IO function, this function remains active until the device is switched off.

For this reason, switch the device off after sending a project without PROFINET IO function. This is required, for example, if you convert the gateway function in the MSI 400 project from PROFINET IO to Modbus TCP.

## 7.7 Status bits

The PROFINET IO gateway MSI-EN-PN sets status bits, which are available in the logic editor of MSI.designer for processing.

Tab. 7.4: Meaning of the state bits MSI-EN-PN[0] in the logic editor

| Name of the state bits | Set to 1, if ... | Reset to 0 ... |
|---|---|---|
| Output status | … Data from slot 16, 17, 18, 19, 20, 32 or 33 was transmitted without error. | … No AR (Application Relation) exists. |
| Input status | … Data from slot 21, 22, 23, 24 or 25 was downloaded from a PLC without error. | … No AR (Application Relation) exists. |
| Internal state | … An AR (Application Relation) is active. | ... No AR exists. |

An Application Relation (AR) is a clear communication relationship between two communication partners, for example a PLC and a device. The AR is initialized during PLC start-up. Cyclical input and output data, acyclical data using read/write services and alarms are exchanged bidirectionally between the PLC and the device within this AR.

## 7.8 Optimizing performance

Only use the data blocks from the hardware catalog of the module that you actually need for your application.

Sequence the process data in the routing tables within a data block without gaps (see *Layout and content of the tabs [chapter 5.1.3]*). Then check whether this will enable you to do without the use of individual data blocks from the hardware catalog. This helps to reduce the number of data bytes periodically exchanged in the network.

# 8 EtherNet/IP gateway

This chapter describes the "EtherNet/IP-Gateway" function of the MSI 430-x/???- LEUZE module.

The EtherNet/IP protocol is not described in this chapter. If you have little or no experience with this, please refer to the ODVA documentation for more information. Some content can be found in the glossary (see *Abbreviations and definitions [chapter 1.5]*).

| | NOTICE |
|---|---|
| | **Use of the term "Device" in this chapter**<br>This chapter uses the term "Device" as a synonym for the controller module MSI 430-x/???-LEUZE. |

## 8.1 Interfaces and operation

Interfaces and operation are identical to that of the Modbus TCP Gateway.

Read the following section: *Interfaces and operation [chapter 6.1]*

## 8.2 Data sheet

The MSI 430-x/???- LEUZE module supports EtherNet/IP from product version D-01.01 onwards. The following functions are integrated:

- Implicit message transmission (transport class 1)
- Explicit message transmission (transport class 3, connected)
- Device profile: Discrete universal I/O device
- UCMM Message Server (no connection)
- Supported objects: Message router, connection manager, port, identity, Ethernet link, TCP/IP, I/O point and group (discrete), vendor class 0x78, assembly
- Up to five simultaneous encapsulation sessions (input and output)
- Assemblies of a variable size
- Supported addressing: Class/instance/attribute and symbol tag
- Agreement with CIP (Common Industrial Protocol) specification and with EtherNet/IP CIP specification, according to table *Module versions and referenced specification versions for EtherNet/IP [chapter 8.2]*
- Details EDC file with ODVA conformity test
- Supported PCCC commands: Read and write word range, read and write input, read and write protected logic input with two and three address fields for connection to PLC 3, PLC 5, PLC 5/250, PLC 5/VME, SLC 500, SLC 5/03, SLC 5/04 and MicroLogix-1000
- Automatic configuration of semi and full duplex connections as well as of connections with 10 and 100 Mbit/s.
- MS (module state) and NET (network) LED

Tab. 8.1: Module versions and referenced specification versions for EtherNet/IP

| Module version | CIP (Common Industrial Protocol) specification | EtherNet/IP CIP specification |
|---|---|---|
| up to D-01 | Version 3.18 | Version 1.19 |
| from D-03 | Version 3.21 | Version 1.22 |

## 8.3 Basic setup

### 8.3.1 Basic configuration of PLC

This chapter briefly describes the basic configuration of the PLC.

Firstly, install the current EDS file for the MSI 430-x/???- LEUZE module in your PLC configuration program. You can find the current EDS file on the Internet at www.leuze-shop.com. The following diagram shows you how you can make the setting using the Logix Designer.

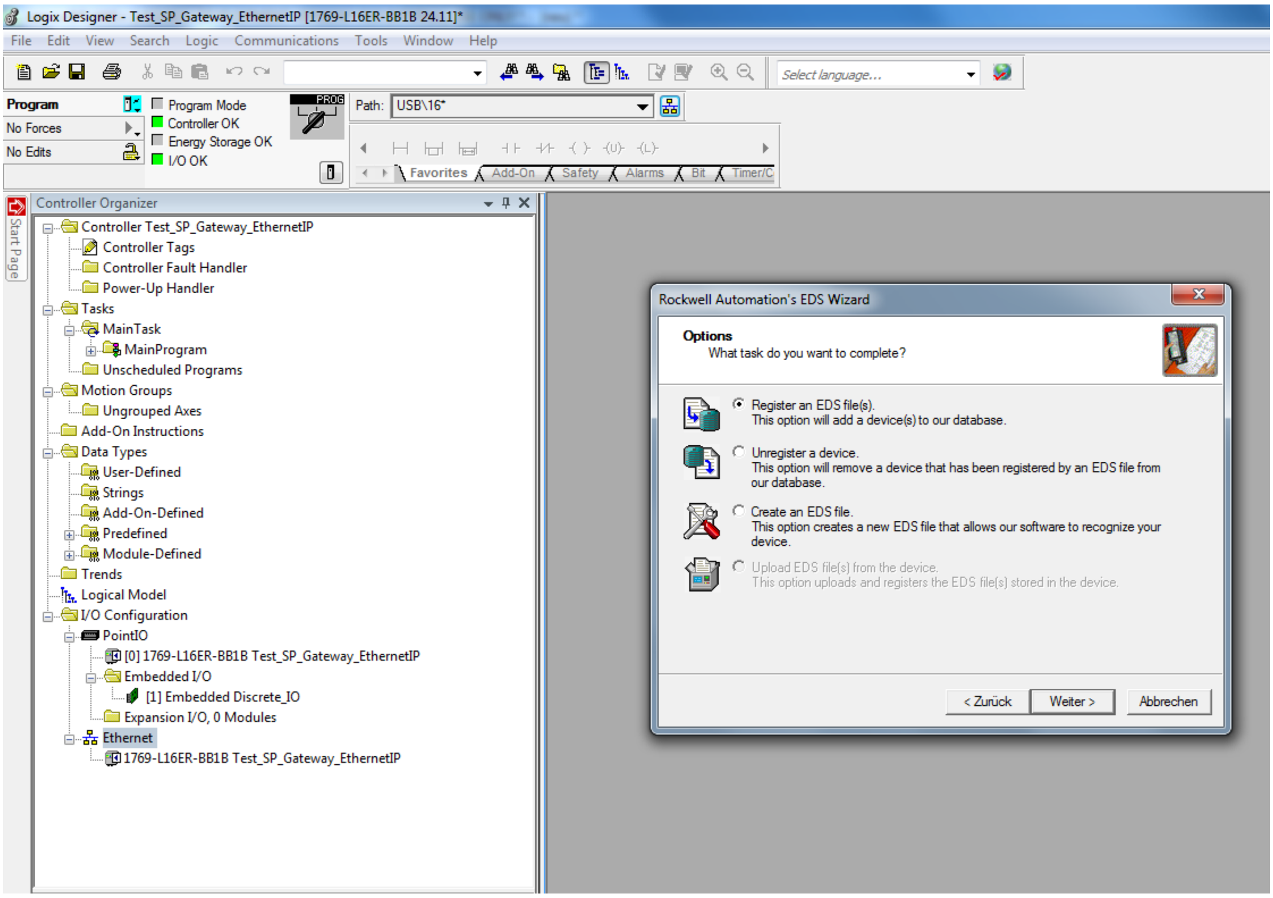



Fig. 8.1: Register the EDS file using the ESD Wizard in the Logix Designer

The article number is "50132988,50132989,50134315,50134316" and can be filtered according to the vendor name "Leuze electronic" or a part of this name.

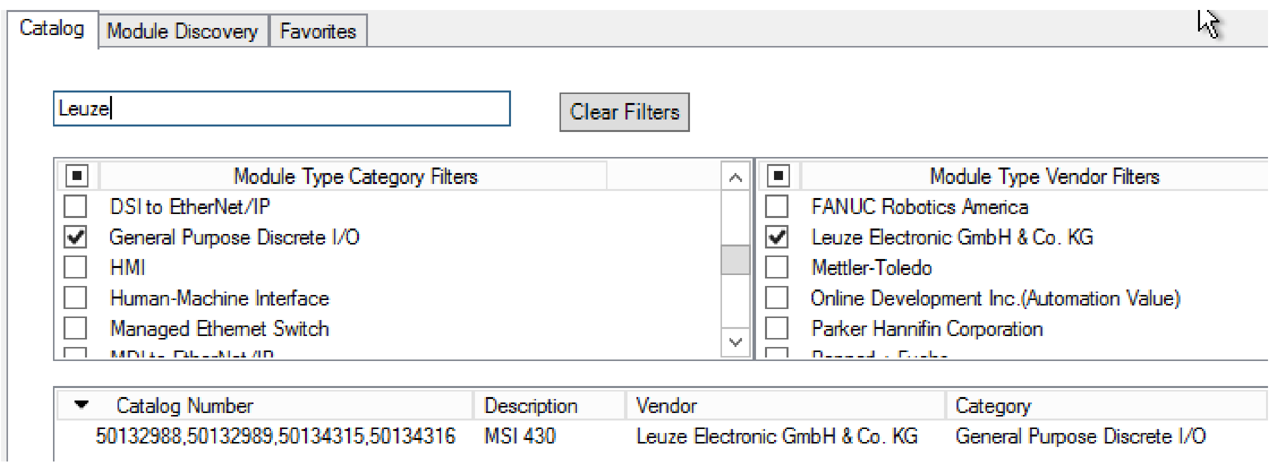



Fig. 8.2: Selection of the module type in the Logix Designer

In the **Internet Protocol** tab in the Logix Designer, select **Manually configure IP settings**. Select the required IPv4 address and the appropriate network mask.

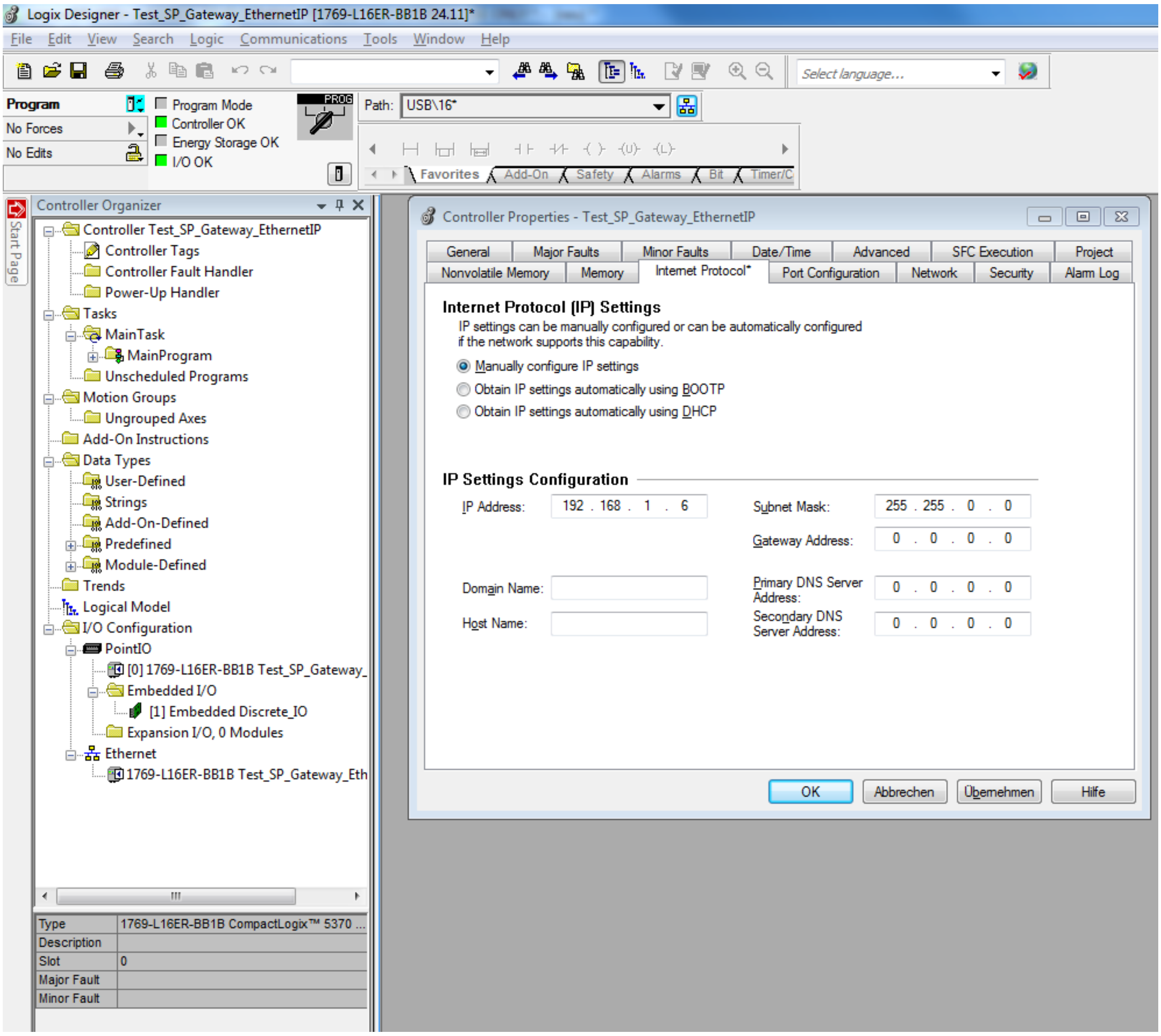



Fig. 8.3: IPv4 setting for the device in the Logix Designer

The MSI 430-x/???- LEUZE module is a **General Purpose Discrete I/O Device.** For quick installation, use the connection **Logic Output (1 to 400) and Logic/Physical Input**, if your PLC supports implicit message transmission. The following figure shows the appropriate dialog in the Logix Designer.

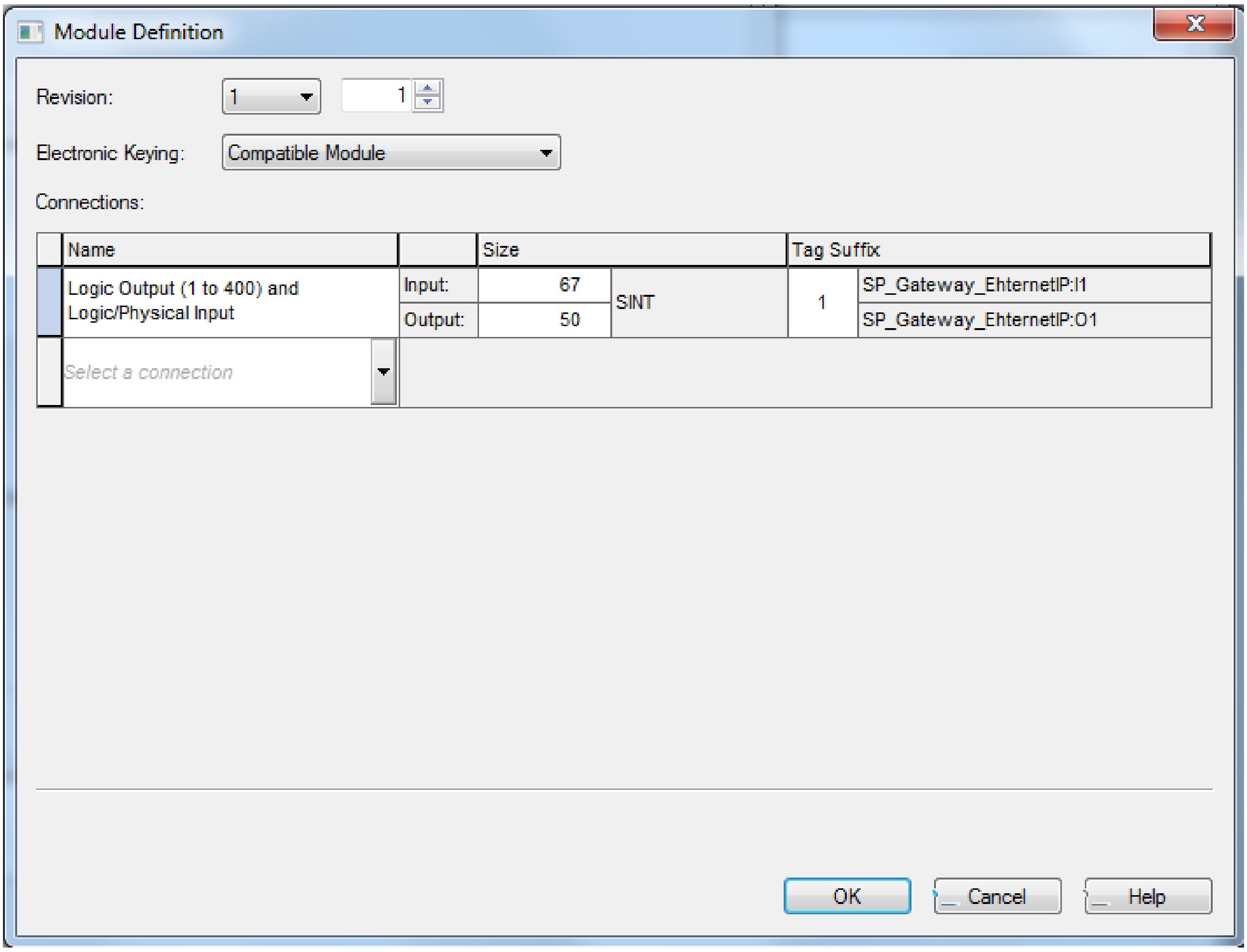



Fig. 8.4: Basic connection selected in the Logix Designer

This connection comprises up to 50 bytes for data transmission from the PLC to the MSI 430-x/???- LEUZE module (assembly instance 37). This connection comprises up to 67 bytes for data transmission from the MSI 430-x/???- LEUZE module to the PLC (assembly instance 57). The following table offers an overview of these data bytes.

Tab. 8.2: Data of the class 1 connection “Logic output (1 to 400) and logic/physical input”

| Instance | Byte | Access | Data type | Description | Size | Data range |
|---|---|---|---|---|---|---|
| 37 | 0 to 49 | Write, read | BYTE[50] | Output bytes, configuration via **Input data block 1 to 5** in MSI.designer<br>(*More [chapter 8.5.2.1]*) | 1 to 50 Bytes | 0 to 0xff |
| 57 | 0 to 49 | Read | BYTE[50] | Input bytes, configuration via **Output data block 1** in MSI.designer<br>(*More [chapter 8.5.3.1]*) | 1 to 50 Bytes | 0 to 0xff |
| | 50 to 65 | Read | BYTE[16] | Bits of the input terminals (instance 401 to 528 of attribute 3 class 8, currently not listed in MSI.designer)<br>(*More [chapter 8.5.3.1]*) | 1 to 16 Bytes | 0 to 0xff |
| | 66 | Read | BYTE | Bit 7: Input state<br>Bit 6: Output state<br>(currently not listed in MSI.designer) | 1 bytes | 0x00, 0x40, 0x80, 0xc0 |

Other connections supported by the MSI 430-x/???- LEUZE module are listed in the following table. You can find information about these assembly instances in the table "*Overview of assembly data bytes [chapter 8.5.1]*".

Tab. 8.3: Class 1 connections supported by the MSI 430-x/???- LEUZE module

| Name of the connection | Assembly for data from the PLC to MSI 4xx (O→T) | Assembly for data from MSI 4xx to the PLC (T→O) |
|---|---|---|
| Logic output (1 to 400) and logic/physical input | 37 | 57 |
| Logic output (1 to 400) and logic/state/system mode assembly | 37 | 167 |
| Logic output (81 to 400) and logic/physical input | 138 | 57 |
| Logic output (81 to 400) and logic/state/system mode assembly | 138 | 167 |
| Logic output (161 to 400) and logic/physical input | 139 | 57 |
| Logic output (161 to 400) and logic/state/system mode assembly | 139 | 167 |
| Logic output (241 to 400) and logic/physical input | 140 | 57 |
| Logic output (241 to 400) and logic/state/system mode assembly | 140 | 167 |
| Logic output (321 to 400) and logic/physical input | 141 | 57 |
| Logic output (321 to 400) and logic/state/system mode assembly | 141 | 167 |
| Logic/physical input (“Listen only”) | 199 | 57 |
| Logic/state/system mode assembly (“Listen only”) | 199 | 167 |
| Logic/physical input (“Input only”) | 198 | 57 |
| Logic/state/system mode assembly (“Input only”) | 198 | 167 |

Connection point 199 (= 0xc7) is used for **Listen Only** and connection point 198 (= 0xc6) for **Input Only**. Both possess a data size of zero. This means that the PLC does not make any data available for the MSI 430-x/???- LEUZE module.

If the PLC only requires process data from the MSI 4xx module, the user is recommended to use a connection with **Input Only**.

### 8.3.2 Basic configuration of the controller module

The integrated gateway MSI-EN-IP (EtherNet/IP gateway) is part of the MSI 430-x/???- LEUZE module.

#### Activating the gateway

You can activate the integrated gateway in the configuration dialog box of the MSI 430-x/???- LEUZE module, in the **Properties** docking window:

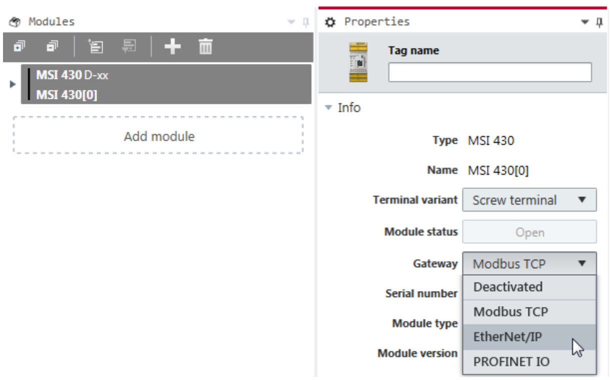



Fig. 8.5: Activation of EtherNet/IP in MSI.designer

| | NOTICE |
|---|---|
| ! | It is possible to select the MSI-EN-IP gateway for modules of type MSI 430-x/???- LEUZE from version D-01.01 onwards. |

**Adapting the IPv4 data**

The IPv4 data of the MSI 430-x/???- LEUZE module can be adapted to the PLC settings in MSI.designer.

**Requirement**

During transmission of the IPv4 data, the device must not be in **Run (Execute)** mode. The command bar must be displayed on the left above the **Start** command, as shown in the following illustration. If this is not the case, stop the device via the **Stop** button.

**Required window layout**

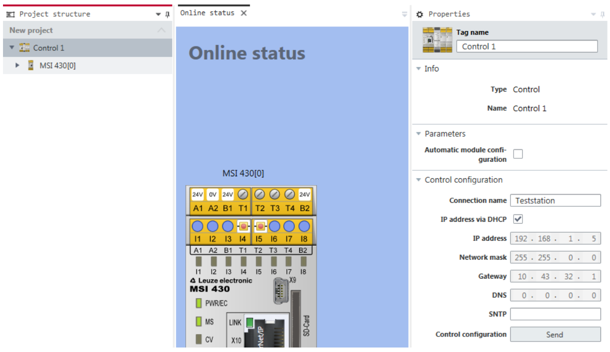



Fig. 8.6: Setting of the IPv4 device data in MSI.designer

## 8.3.3 Configuring the data to the PLC

The data transferred to the PLC and thus from the target device to the sender (Target to Originator, T→O) can be adapted in the "MSI 400 → MSI-EN-IP[0]" tab of the gateway configuration in MSI.designer. By default, the first three bytes contain data for the input terminals I1 to I16 (and IQ1 to IQ4 in the appropriate configuration as an input). Byte 4 comprises data of the output terminals Q1 to A4 (and IQ1 to IQ4 in the appropriate configuration as an output).

Bytes 12 to 23 comprise data for the input terminals I1 to I8 of the input/output expansion modules. Bytes 24 to 35 comprise data for the output terminals Q1 to Q4 of the expansion modules MSI-EM-IO84 or MSI-EM-IO84NP. Bytes 4 to 11 comprise data of the logic editor and are called **Direct Off**.

This standard configuration can be adapted as shown here using drag & drop from the **Gateway** docking window in the tabs for the gateway configuration:

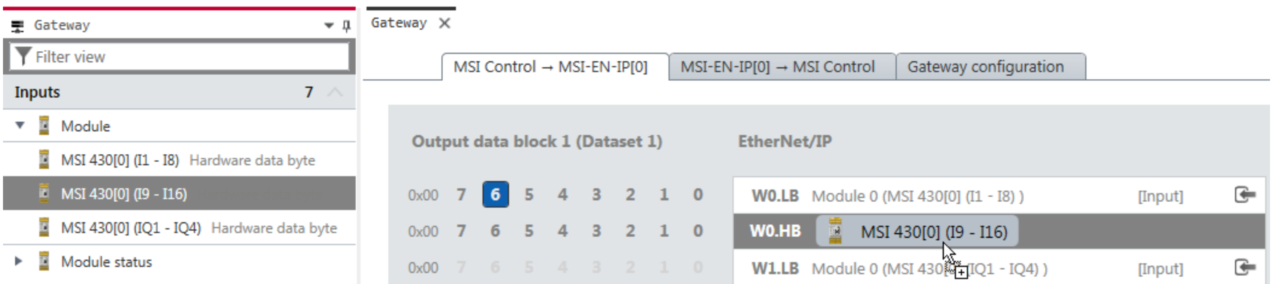



Fig. 8.7: Adding of bytes to the gateway process image (T→O) using drag-and-drop in MSI.designer

In addition, the tag names of all the bytes in MSI.designer can be added or edited, in order to be able to use them in the **Logic** view of MSI.designer. User-defined names improve program legibility and troubleshooting. Tag names can be configured in the **Parameters** section of the **Properties** docking window.

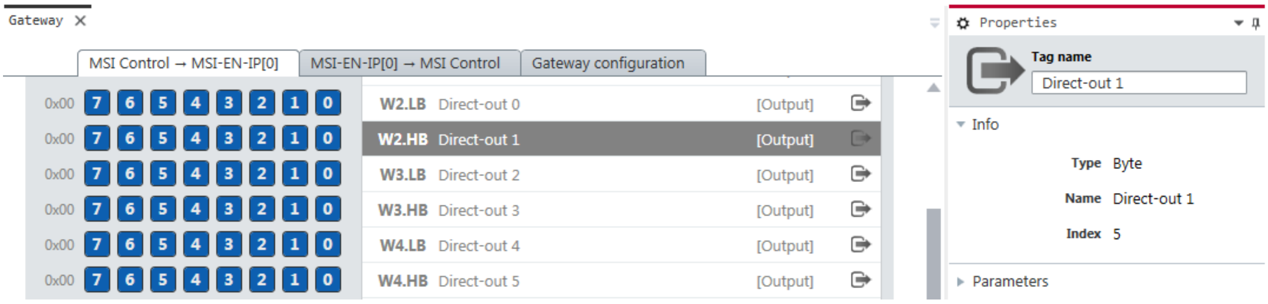



Fig. 8.8: Adding new data bytes (T→O) for use in the "Logic" view by configuring tag names

## 8.3.4 Configuring the data from the PLC

Data transmitted by the PLC and thus by the sender to the target device (Originator to Target, O→T) can be named in the "MSI-EN-IP [0] → MSI 400" tab for the gateway configuration in MSI.designer. By default, the logic values **Direct On 0** to **Direct On 3** are assigned to the four first bytes. The names **Data bit 0** to **Data bit 7** are assigned to each bit as standard. Each bit can be used in the **Logic** view of MSI.designer as an unsafe input element, such as a Restart button or as a signal lamp.

Additional input elements for gateway data can be added as necessary by the configuration of additional tag names.

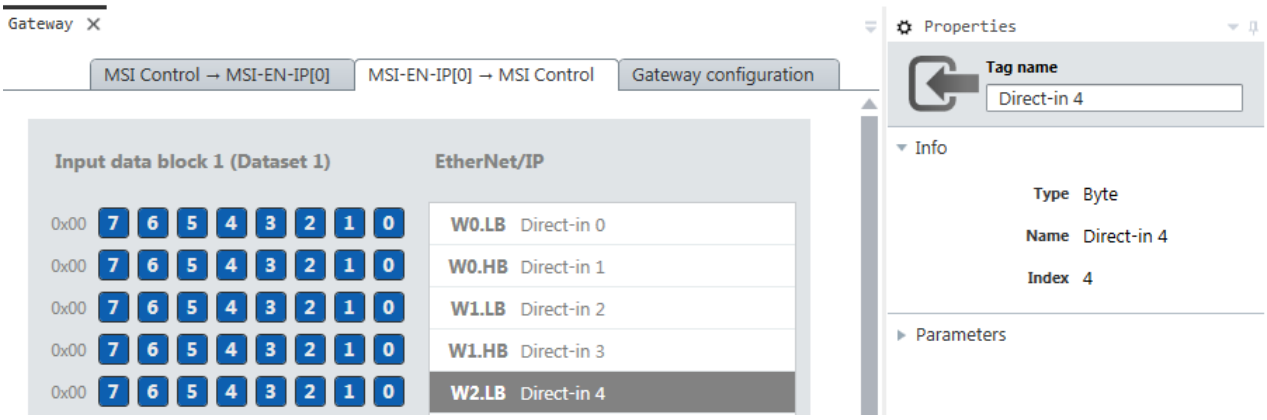



Fig. 8.9: Adding of a new data byte (T→O) for use in the logic editor by configuring the tag name.

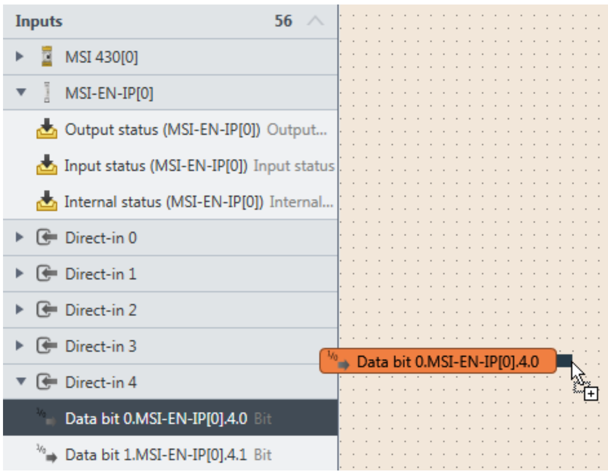



Fig. 8.10: Adding gateway data elements to the logic editor in MSI.designer via drag & drop

## 8.4 Supported CIP objects

### 8.4.1 Identity object

The identity object is required for all EtherNet/IP-based products. Instance 1, attribute 1 stands for the Vendor ID. Leuze electronic GmbH is listed by the ODVA using the value 524.

Instance 1, attribute 2 stands for the device types. The Open Type Code 0x07 stands for a **discrete universal I/O device.**

Instance 1, attribute 3 stands for the product code. It is of the type UNIT and thus comprises 2 bytes.

Instance 1, attribute 4 stands for the revision, that means the main and supplementary firmware version of the MSI 430-x/???- LEUZE module, which you can find in the MSI.designer software as the **Diagnostics version**. You can see both details in the **Properties** docking window, if you select the controller module in the **Modules** docking window after you have connected to the station.

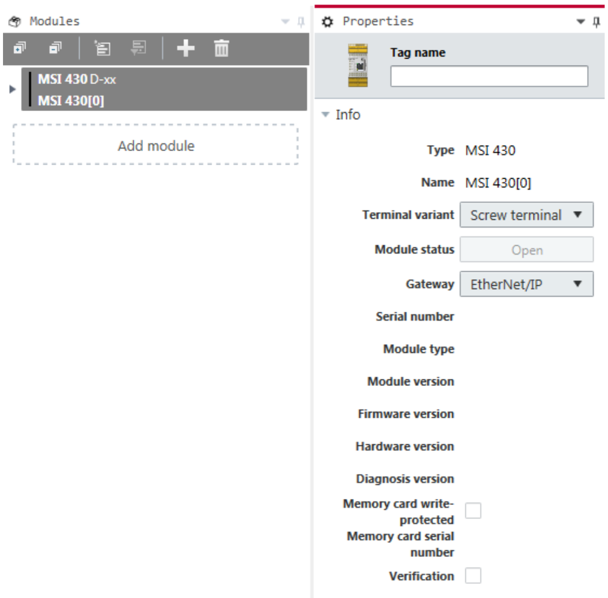



Fig. 8.11: Display of diagnostic version and hardware version in configuration dialog of the controller module

Instance 1, attribute 5 standard for the current state of the overall device. The data range is listed in the *Device state values table in class 1, instance 1, attribute 5 [chapter 8.4.1]*.

Instance 1, attribute 6 stands for the serial number of the device, which can be found under the hardware configuration in MSI.designer. Instance 1, attribute 7 stands for the product name MSI 430-x/???- LEUZE.

Tab. 8.4: Overview of the identity class (0x01) supported by the MSI 430-x/???- LEUZE module

| Class | Instance | Attribute | Access | Data type | Description | Data range |
|---|---|---|---|---|---|---|
| 1 | 0 = Class | 1 | Read | UINT | Revision | 1 |
| 1 | 0 = Class | 2 | Read | UINT | Max. instance | 1 |
| 1 | 0 = Class | 3 | Read | UINT | Number of instances | 1 |
| 1 | 0 = Class | 6 | Read | UINT | Max. class attribute ID | 7 |
| 1 | 0 = Class | 7 | Read | UINT | Max. instance attribute ID | 7 |
| 1 | 1 | 1 | Read | UINT | Vendor ID | 524 = 0x20c |
| 1 | 1 | 2 | Read | UINT | Device type | 0x07 |
| 1 | 1 | 3 | Read | UINT | *Product code [chapter 8.4.8.3]* | 05001 |
| 1 | 1 | 4 | Read | USINT[2] | Revision, software version<br>The “left” byte is the main section and is transmitted first | {1, 1} to {99, 99} |
| 1 | 1 | 5 | Read | WORD | Device state | See next table |
| 1 | 1 | 6 | Read | UDINT | Serial number | 16010001 to 99539999 |
| 1 | 1 | 7 | Read | SHORT_ STRING | Product name | MSI 430-x/???- LEUZE |

Tab. 8.5: Device status values of the MSI 430-x/???- LEUZE module in class 1, instance 1, attribute 5

| State value | Description | Possible system mode |
|---|---|---|
| 0b0000 xxxx xxxx 0x01 | There is at least one EtherNet/IP connection to a PLC (owner of the connection) | 4 = Idle<br>5 = Run<br>7 = Critical error<br>21 = Force mode |
| 0b0000 xxxx 0000 010x | Device is configured | 4 = Idle<br>5 = Run<br>7 = Critical error<br>21 = Force mode |
| 0b0000 0001 0000 0x0x | Low, removable error | 4 = Idle<br>5 = Run<br>21 = Force mode |
| 0b0000 0010 0000 0x0x | Low, non-removable error | 4 = Idle<br>5 = Run<br>21 = Force mode |
| 0b0000 0100 0000 0x0x | Serious, removable error | 1 = Init<br>2 = Configuration required<br>3 = Configuration running |
| 0b0000 1000 0000 0x0x | Serious, non-removable error | 7 = Critical error |

### 8.4.2 Assembly object

All the data of the Class 1 connections are also provided by the Assembly object. The following table offers an overview of this assembly object.

Further information:

- Table *Overview of assembly data bytes of the MSI 430-x/???- LEUZE module [chapter 8.5.1]*
- Figure *Data flow when using assembly instances [chapter 8.6.2]*
  (Shows the data flow upstream of the PLC to the MSI 430-x/???- LEUZE module and back from the point of view of the individual assemblies.)

Tab. 8.6: Overview of the assembly class (0x04) supported by the MSI 430-x/???- LEUZE module

| Class | Instance | Attribute | Access | Data type | Description | Data range |
|---|---|---|---|---|---|---|
| 4 | 0 = Class | 1 | Read | UINT | Revision of the class | 2 |
| 4 | 0 = Class | 2 | Read | UINT | Max. instance | 167 |
| 4 | 0 = Class | 3 | Read | UINT | Number of instances | 7 |
| 4 | 0 = Class | 6 | Read | UINT | Max. class attribute ID | 7 |
| 4 | 0 = Class | 7 | Read | UINT | Max. instance attribute ID | 4 |
| 4 | 37 | 1 | Read | UINT | Number of members | 0 |
| 4 | 37 | 3 | Read, write | BYTE[50] | *Bits of the logic outputs [chapter 8.5.2]* (Instance 1 to 400 of Class 9) | See [1)] |
| 4 | 37 | 4 | Read | UINT | Number of data bytes | 50 |
| 4 | 57 | 1 | Read | UINT | Number of members | 0 |
| 4 | 57 | 3 | Read | BYTE[67] | Input bits (Instance 1 to 528 of Class 8) | See [1)] |
| 4 | 57 | 4 | Read | UINT | Number of data bytes | 67 |
| 4 | 138 | 1 | Read | UINT | Number of members | 0 |
| 4 | 138 | 3 | Read, write | BYTE[40] | *Bits of the logic outputs [chapter 8.5.2]* (Instance 81 to 400 of Class 9) | See [1)] |
| 4 | 138 | 4 | Read | UINT | Number of data bytes | 40 |
| 4 | 139 | 1 | Read | UINT | Number of members | 0 |
| 4 | 139 | 3 | Read, write | BYTE[30] | *Bits of the logic outputs [chapter 8.5.2]* (Instance 161 to 400 of Class 9) | See [1)] |
| 4 | 139 | 4 | Read | UINT | Number of data bytes | 30 |
| 4 | 140 | 1 | Read | UINT | Number of members | 0 |
| 4 | 140 | 3 | Read, write | BYTE[20] | *Bits of the logic outputs [chapter 8.5.2]* (Instance 241 to 400 of Class 9) | See [1)] |
| 4 | 140 | 4 | Read | UINT | Number of data bytes | 20 |
| 4 | 141 | 1 | Read | UINT | Number of members | 0 |
| 4 | 141 | 3 | Read, write | BYTE[10] | *Bits of the logic outputs [chapter 8.5.2]* (Instance 321 to 400 of Class 9) | See [1)] |
| 4 | 141 | 4 | Read | UINT | Number of data bytes | 10 |
| 4 | 167 | 1 | Read | UINT | Number of members | 0 |

| Class | Instance | Attribute | Access | Data type | Description | Data range |
|---|---|---|---|---|---|---|
| 4 | 167 | 3 | Read | BYTE[112] | Bits of the logic inputs, mode and state bytes<br>(*More [chapter 8.5.3.2]*) | See [1)] |
| 4 | 167 | 4 | Read | UINT | Number of data bytes | 112 |
| [1)]See: Table *Overview of assembly data bytes of the MSI 430-x/???- LEUZE module [chapter 8.5.1]* | | | | | | |

### 8.4.3 Discrete input point objects

The discrete input point objects are part of the device profile **Discrete universal I/O device.**

If an error occurs at the terminal input of a specific instance between 401 and 528 and the MSI 430-x/???- LEUZE module is in **Run** mode, the value of the instance attribute 4 equals 1. In all other cases, the value equals 0.

Tab. 8.7: Overview of the discrete input point objects (0x08) supported by the MSI 430-x/???- LEUZE module

| Class | Instance | At-tribute | Access | Data type | Description | Data range |
|---|---|---|---|---|---|---|
| 8 | 0 = Class | 1 | Read | UINT | Revision of the class | 2 |
| 8 | 0 = Class | 2 | Read | UINT | Max. instance | 584 |
| 8 | 0 = Class | 3 | Read | UINT | Number of instances | 400 + 128 + 56<br>Logic + input + output |
| 8 | 0 = Class | 6 | Read | UINT | Max. class attribute ID | 7 |
| 8 | 0 = Class | 7 | Read | UINT | Max. instance attribute ID | 4 |
| 8 | 1 to 400 and 529 to 584 | 1 | Read | USINT | Number of attributes | 3 |
| 8 | 401 to 528 | 1 | Read | USINT | Number of attributes | 4 |
| 8 | 1 to 528 | 2 | Read | USINT[4] | List of support attributes | {1, 2, 3, 4} |
| 8 | 529 to 584 | 2 | Read | USINT[3] | List of support attributes | {1, 2, 3} |
| 8 | 1 to 400 | 3 | Read | BOOL | The value of the input bit, configured by the **output data set 1** in MSI.designer, stands for the data transferred by the logic of the controller module to the PLC. | 0 = Off,<br>1 = On |
| 8 | 1 to 400 | 4 | Read | BOOL | State of **output data set 1** | 0 = OK |
| 8 | 401 to 416 | 3 | Read | BOOL | Value of terminals I1 to I16 of the MSI 430-x/???- LEUZE module | 0, 1 |
| 8 | 401 to 416 | 4 | Read | BOOL | Status of terminals I1 to I16 of the MSI 430-x/???- LEUZE module | 0, 1 |
| 8 | 417 to 420 | 3 | Read | BOOL | Value of terminals IQ1 to IQ4 of the MSI 430-x/???- LEUZE module when configured as an input | 0, 1 |
| 8 | 417 to 420 | 4 | Read | BOOL | Status of terminals IQ1 to IQ4 of the MSI 430-x/???- LEUZE module when configured as an input | 0, 1 |
| 8 | 421 to 430 | 3 | Read | BOOL | Reserved | 0 |

| Class | Instance | At-<br>tribute | Access | Data type | Description | Data range |
|---|---|---|---|---|---|---|
| 8 | 431 | 3 | Read | BOOL | Value of B1 | Voltage is...<br>0 = Outside the tolerance<br>1 = Within the tolerance |
| 8 | 432 | 3 | Read | BOOL | Value of B2 | Voltage is...<br>0 = Outside the tolerance<br>1 = Within the tolerance |
| 8 | 421 to 432 | 4 | Read | BOOL | Reserved | 0 |
| 8 | 425 + 8 x n to<br>432 + 8 x n | 3 | Read | BOOL | Value of terminals I1 to I8 of the MSI-EM-I8[n] / ???- LEUZE[n] module,<br>with n = 1 to 12 | 0, 1 |
| 8 | 425 + 8 x n to<br>432 + 8 x n = 528 | 4 | Read | BOOL | State of terminals I1 to I8 of the MSI-EM-I8[n] / ???- LEUZE[n] module,<br>where n = 1 to 12 | 0, 1 |
| 8 | 529 to 532 | 3 | Read | BOOL | Value of terminals Q1 to Q4 of the MSI 430-x/???- LEUZE module | 0, 1 |
| 8 | 533 to 536 | 3 | Read | BOOL | Value of terminals IQ1 to IQ4 of the MSI 430-x/???- LEUZE module when configured as an output | 0, 1 |
| 8 | 533 + 4 x n to<br>536 + 4 x n = 584 | 3 | Read | BOOL | Value of terminals Q1 to Q4 of the MSI-EM-IO84[n] module,<br>where n = 1 to 12 | 0, 1 |

### 8.4.4 Discrete output point objects

The discrete output point objects are part of the device profile **Discrete universal I/O device.**

The MSI 400 system does not permit direct influencing of the security-oriented output terminals. Instead, up to 400 databits can be specified. In this way, it is possible to use the **input data blocks 1 to 5** in MSI.designer for bit-wise access. The simplest way to control output terminals with a PLC is by connecting the appropriate gateway bit to an output in the logic editor of MSI.designer. The following figure shows an example:

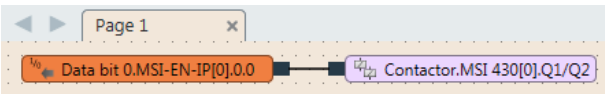



Fig. 8.12: Direct connection of a gateway input bit to an output terminal of the MSI 430-x/???- LEUZE module

**WARNING**

**Check your application thoroughly for correctness!**

Because the MSI.designer only checks for logic-internal connection errors, you have to check the following aspects systematically yourself:

- Does your application correspond to the results from the risk analysis and the avoidance strategy?
- Have all of the applicable standards and guidelines been complied with?
  If not, you are placing the machine operator in danger.

Note that the output terminal is set to **Off** as standard and thus stands for the value “0”. This value is always used when the controller module is not in **Run** mode or it the output is not configured via the logic editor in MSI.designer.

The standard value of gateway output bits can be configured using attributes 5 and 6.

If there is a loss of connection between the PLC and the controller module, instance attribute 5 controls whether the gateway data bit is set or not. The specified value is controlled by instance attribute 6.

A write request to attribute 3 of instances 1 to 400 is refused if the *Assembly instance 37 [chapter 8.5.2.1]* is already linked to an active connection to a PLC.

A write request to attribute 3 of instances 81 to 400 is refused if the *Assembly instance 138 [chapter 8.5.2]* is already linked to an active connection to a PLC.

A write request to attribute 3 of instances 161 to 400 is refused if the *Assembly instance 139 [chapter 8.5.2]* is already linked to an active connection to a PLC.

A write request to attribute 3 of instances 241 to 400 is refused if the *Assembly instance 140 [chapter 8.5.2]* is already linked to an active connection to a PLC.

A write request to attribute 3 of instances 321 to 400 is refused if the *Assembly instance 141 [chapter 8.5.2]* is already linked to an active connection to a PLC.

Tab. 8.8: Overview of the discrete output point objects (0x09) supported by the MSI 430-x/???- LEUZE module

| Class | Instance | Attribute | Access | Data type | Description | Data range |
|---|---|---|---|---|---|---|
| 9 | 0 = Class | 1 | Read | UINT | Revision of the class | 1 |
| 9 | 0 = Class | 2 | Read | UINT | Max. instance | 400 |
| 9 | 0 = Class | 3 | Read | UINT | Number of instances | 400 |
| 9 | 0 = Class | 6 | Read | UINT | Max. class attribute ID | 7 |
| 9 | 0 = Class | 7 | Read | UINT | Max. instance attribute ID | 6 |
| 9 | 1 to 400 | 1 | Read | USINT | Number of attributes | 5 |
| 9 | 1 to 400 | 2 | Read | USINT[5] | List of support attributes | {1, 2, 3, 5, 6} |
| 9 | 1 to 400 | 3 | Write, read | BOOL | The value of the logic output bit, which is configured by the **intput data blocks 1 to 5** in MSI.designer, stands for the data transferred by the PLC to the logic of the controller module. | 0 = Off, 1 = On |
| 9 | 1 to 400 | 5 | Write, read | BOOL | Error action (specified value on loss of connection to the PLC) | 0 = Interference value, 1 = Last state |
| 9 | 1 to 400 | 6 | Write, read | BOOL | Interference value | 0 = Off, 1 = On |

### 8.4.5 Discrete input group object

The discrete input group objects are part of the device profile **Discrete universal I/O device.**

The object of class 29 plays a role with regard to the alarm bit. It collects the process alarms of all the input terminals of the MSI 430-x/???- LEUZE module as well as the safe input/output expansion modules in one bit. If an error occurs in at least one input terminal and the MSI 430-x/???- LEUZE module is in **Run** mode, the value of the attribute 5 of instance 1 equals 1. In all other cases, the value equals 0.

Tab. 8.9: Overview of the discrete input group object (0x1D) supported by the MSI 430-x/???- LEUZE module

| Class | Instance | Attribute | Access | Data type | Description | Data range |
|---|---|---|---|---|---|---|
| 29 | 0 = Class | 1 | Read | UINT | Revision of the class | 1 |
| 29 | 0 = Class | 2 | Read | UINT | Max. instance | 1 |
| 29 | 0 = Class | 3 | Read | UINT | Number of instances | 1 |
| 29 | 0 = Class | 6 | Read | UINT | Max. class attribute ID | 7 |
| 29 | 0 = Class | 7 | Read | UINT | Max. instance attribute ID | 5 |
| 29 | 1 | 1 | Read | USINT | Number of attributes | 5 |
| 29 | 1 | 2 | Read | USINT[5] | List of support attributes | {1, 2, 3, 4, 5} |
| 29 | 1 | 5 | Read | BOOL | Group state of all input terminals (state of instances 401 to 420 of class 8) | 0 = No error, 1 = Error |

### 8.4.6 Discrete output group object

The discrete output group objects are part of the device profile **Discrete universal I/O device.**

The object of class 30 plays a role with regard to the alarm bit. It collects the process alarms of all the output terminals of a MSI 430-x/???- LEUZE or MSI-EM-IO84 module in one bit. If an error occurs in at least one output terminal and the MSI 430-x/???- LEUZE module is in **Run** mode, the value of the attribute 5 of instance 1 is 1. If the MSI 430-x/???- LEUZE module is in **Critical error** mode, the attribute value is also 1. In all other cases, the value equals 0.

Tab. 8.10: Overview of the discrete output group object (0x1D) supported by the MSI 430-x/???- LEUZE module

| Class | Instance | Attribute | Access | Data type | Description | Data range |
|---|---|---|---|---|---|---|
| 30 | 0 = Class | 1 | Read | UINT | Revision of the class | 1 |
| 30 | 0 = Class | 2 | Read | UINT | Max. instance | 1 |
| 30 | 0 = Class | 3 | Read | UINT | Number of instances | 1 |
| 30 | 0 = Class | 6 | Read | UINT | Max. class attribute ID | 7 |
| 30 | 0 = Class | 7 | Read | UINT | Max. instance attribute ID | 6 |
| 30 | 1 | 1 | Read | USINT | Number of attributes | 6 |
| 30 | 1 | 2 | Read | USINT[6] | List of support attributes | {1, 2, 3, 4, 5, 6} |
| 30 | 1 | 3 | Read | USINT | Number of bound instances | 56 |
| 30 | 1 | 4 | Read | UINT[56] | Bound instances | {1, ..., 56} |
| 30 | 1 | 5 | Read | BOOL | Group state of all output terminals (state of instances 529 to 584 of class 8) | 0 = No error, 1 = Error |

### 8.4.7 PCCC object

PCCC (pronounced “P C Cube”) is used in several PLCs from Rockwell Automation/Allen Bradley, which still continue to be used. It was developed before CIP and EtherNet/IP were defined. PCCC telegrams are either:

a) Encapsulated in CIP packages (e.g. via EtherNet/IP)

b) The encapsulation of CIP packages.

The MSI 430-x/???- LEUZE module supports the encapsulation of PCCC data in CIP packages, as described under b) above. For this, the class ID 0x67 = 103 was specified.

The PCCC commands listed in the following table are supported by the MSI 430-x/???- LEUZE module.

All PCCC-related data with a size of 16 bits (word) are available in the “Little Endian” format. This means that the byte with the lowest value is executed first.

Tab. 8.11: PCCC commands supported by the MSI 430-x/???- LEUZE module

| Type | CMD | FNC | Description | Command supported by |
|---|---|---|---|---|
| PLC-5 | 0x0f | 0x00 | *Write word range [chapter 8.4.7.2]* | PLC-3, PLC-5, PLC-5/250 |
| PLC-5 | 0x0f | 0x01 | *Read word range [chapter 8.4.7.3]* | PLC-3, PLC-5, PLC-5/250 |
| PLC-5 | 0x0f | 0x67 | *Write input [chapter 8.4.7.4]* | SLC 5/03, SLC 5/04, PLC 5, PLC-5/250, PLC-5/VME |
| PLC-5 | 0x0f | 0x68 | *Read input [chapter 8.4.7.5]* | SLC 5/03, SLC 5/04, PLC 5, PLC-5/250, PLC-5/VME |
| SLC | 0x0f | 0xa1 | *Read protected logic input with two address fields [chapter 8.4.7.6]* | |
| SLC | 0x0f | 0xa2 | *Read protected logic input with three address fields [chapter 8.4.7.8]* | MicroLogix-1000, SLC 500, SLC 5/03, SLC 5/04, PLC 5 |
| SLC | 0x0f | 0xa9 | *Write protected logic input with two address fields [chapter 8.4.7.7]* | |
| SLC | 0x0f | 0xaa | *Write protected logic input with three address fields [chapter 8.4.7.9]* | MicroLogix-1000, SLC 500, SLC 5/03, SLC 5/04 |

#### 8.4.7.1 PCCC telegram structure

Each request telegram comprises 7+5 header bytes.

Tab. 8.12: PCCC request header

| Name | Data type | Description | Size | Data range |
|---|---|---|---|---|
| Length | USINT | Header size | 1 bytes | 7 |
| Vendor | UINT | Vendor ID of the requester | 2 bytes | |
| S/N | UDINT | Serial number of the requester | 4 bytes | 0 to $2^{32}$-1 |
| CMD | USINT | Command | 1 bytes | 0x0f |
| STS | USINT | State | 1 bytes | 0 |
| TNSW | UINT | Transport sequence number | 2 bytes | 1 to 65535 |
| FNC | USINT | Function code | 1 bytes | 0x67, 0x68, 0xa2, 0xaa |

Each answer telegram comprises 7+4 header bytes or 7+4+1 header bytes, if the state byte is 0xf0.

Tab. 8.13: PCCC reply header

| Name | Data type | Description | Size | Data range |
|---|---|---|---|---|
| Length | USINT | Header size | 1 bytes | 7 |
| Vendor | UINT | Vendor ID of the requester | 2 bytes | |
| S/N | UDINT | Serial number of the requester | 4 bytes | 0 to $2^{32}$-1 |
| CMD | USINT | Command of requester plus Bit 6 set | 1 bytes | 0x4f |
| STS | USINT | State | 1 bytes | 0x00, 0x10, 0xf0 |
| TNSW | UINT | Transport sequence number | 2 bytes | 1 to 65535 |
| EXT STS | USINT | Extended status, only present if STS = 0xf0 | 0 to 1 Bytes | |

### 8.4.7.2 Write word range

The MSI 430-x/???- LEUZE module supports “Write PLC-5 word range” according to the following table:

Tab. 8.14: Data structure of Write PLC-5 word range

| Name | Data type | Description | Data range |
|---|---|---|---|
| Packet offset | UINT | Offset as number of elements | |
| Total Transaction | UINT | Number of elements in the transaction | |
| Address | BYTE[m] | PLC-5 system address, m >= 2 | |
| Payload | UINT[n] | 2•n = Number of data bytes | 0 to 65535 |

The answer of the MSI 430-x/???- LEUZE module does not contain any data, only a status.

### 8.4.7.3 Read word range

The MSI 430-x/???- LEUZE module supports “Read PLC-5 word range” according to the following table:

Tab. 8.15: Read request data structure of PLC-5 word range

| Name | Data type | Description | Data range |
|---|---|---|---|
| Packet offset | UINT | Offset as number of elements | |
| Total Transaction | UINT | Number of elements in the transaction | 0 to value dependent on the assembly size |
| Address | BYTE[m] | PLC-5 system address, m >= 2 | “0” to “:”, “A” to “Z”, “a” to “z” |
| Size | UINT | Number of elements to be returned | |

Tab. 8.16: Feedback to the MSI 430-x/???- LEUZE module of Read PLC-5 word range

| Name | Data type | Description | Data range |
|---|---|---|---|
| Payload | UINT[n] | 2•n = number of data bytes (up to 244 bytes) | 0 to 65535 |

#### 8.4.7.4 Write input

The MSI 430-x/???- LEUZE module supports “Write PLC-5 input” according to the following table:

Tab. 8.17: Write data structure of PLC-5 input

| Name | Data type | Description | Data range |
|---|---|---|---|
| Packet offset | UINT | Offset as number of elements | |
| Total Transaction | UINT | Number of elements in the transaction | |
| Address | BYTE[m] | PLC-5 system address, m >= 2 | See next table |
| Type ID | BYTE[n] | Data type and size, n >= 1 | |

The answer of the MSI 430-x/???- LEUZE module does not contain any data, only a status, see table *PCCC reply header [chapter 8.4.7.1]*. The UINT data format corresponds to writing the format of the word range.

Tab. 8.18: Write address structure of PLC-5 input

| Address | Data type | Number of elements | Description | Data range |
|---|---|---|---|---|
| $N37:x | UINT[n] | n | Output assembly of the device profile Discrete universal I/O device, x = 0 to 24, n = 25 - x | 0 to 65535 |
| $N138:x | UINT[n] | n | Output assembly of the logic output, configured via the **input data block 2 to 5** in MSI.designer,<br>x = 0 to 19, n = 20 – x | 0 to 65535 |
| $N139:x | UINT[n] | n | Output assembly of the logic output, configured via the **input data block 3 to 5** in MSI.designer,<br>x = 0 to 14, n = 15 – x | 0 to 65535 |
| $N140:x | UINT[n] | n | Output assembly of the logic output, configured via the **input data block 4 to 5** in MSI.designer,<br>x = 0 to 9, n = 10 – x | 0 to 65535 |
| $N141:x | UINT[n] | n | Output assembly of the logic output, configured via the **input data block 5** in MSI.designer,<br>x = 0 to 4, n = 5 – x | 0 to 65535 |
| The data range of the number of elements is relative to the assembly sizes. See the table *Overview of assembly data bytes of the MSI 430-x/???- LEUZE module [chapter 8.5.1]* | | | | |

#### 8.4.7.5 Read input

The MSI 430-x/???- LEUZE module supports “Read PLC-5 input” according to the following table:

Tab. 8.19: Read request data structure of PLC-5 word range

| Name | Data type | Description | Data range |
|---|---|---|---|
| Packet offset | UINT | Offset in number of elements | |
| Total Transaction | UINT | Number of elements in the transaction | 0 to value dependent on the assembly size |
| Address | BYTE[m] | PLC-5 system address, m >= 2 | “0” to “:”, “A” to “Z”, “a” to “z” |
| Size | UINT | Number of elements to be returned | |

The answer of the MSI 430-x/???- LEUZE module is listed in the following table. The first byte of the type ID is 0x9a = 0b1001 1010, meaning that the data type is given in the following byte and the data size in the byte after that. The fourth byte of the type ID is 0x42 = 0b0100 0010, standing for an integer data type of size 2.

Tab. 8.20: Feedback to the MSI 430-x/???- LEUZE module for reading the data structure of the PLC-5 input

| Name | Data type | Description | Data range |
|---|---|---|---|
| Type ID | BYTE | Data type and size | Bit 0 to 3: 10 = Size specification in the next but one byte<br>Bit 4 to 7: 9 = Type in the next byte |
| Type ID | BYTE | Data type | 9 = Field of the same elements |
| Type ID | BYTE | Number of following bytes | 1 to n+1 |
| Type ID | BYTE | Data type and size | Bit 0 to 3: 2 = UINT<br>Bit 4 to 7: 4 = Integer |
| Payload | UINT[n] | 2·n = Number of data bytes | 0 to 65535 |

The command data of all assembly instances can be recorded using “Read input”.

In contrast to native addressing of EtherNet/IP assembly instances, the PLC-5 system address contains an element offset which can be used.

The MSI 430-x/???- LEUZE module supports fields (arrays) of UINT as PCCC data types. Due to the odd size of the assembly instance 57, the firmware contained in the MSI 430-x/???- LEUZE module assigns an additional byte, to provide an even number of bytes.

The address scheme supported by the MSI 430-x/???- LEUZE module for Read PLC-5 input is shown in the following table:

Tab. 8.21: Read address structure of PLC-5 input

| Address | Data type | Number of elements | Description | Data range |
|---|---|---|---|---|
| $N57:x | UINT[n] | n | Input assembly of the device profile Discrete I/O device ,<br>x = 0 to 33, n = 34 - x | Element 1 to 33:<br>0 to 65535<br>Element 34 Bit 0 to 7 (LSB): 0x00, 0x40, 0x80, 0xc0<br>Element 34 Bit 8 to 15 (MSB): 0 |
| $N167:x | UINT[n] | n | Input assembly of: | |
| | | | Logic input bits<br>(n = 1-x to 25-x, x = 0 to 24) | 0 to 65535 |
| | | | System state and system mode<br>(n = 26-x, x = 0 to 25) | Bit 0 to 7 (LSB):<br>System mode (1, 2, 3, 4, 5, 7, 21)<br>Bit 8 to 15 (MSB): System state (0x00, 0x40, 0x80, 0xc0) |
| | | | State bytes of the controller module<br>(n = 27-x to 56-x, x = 26 to 55) | 0 to 65535 |

Example: “$N57:10” and “Total Transaction = 24” address elements 11 to 34 correspond to bytes 20 to 66 of assembly instance 57.

Note: Byte 67, which is not specified in assembly instance 57, is also transferred.

Note: The position of the word data with system state and system mode are dependent on the requested amount of data “x”.

#### 8.4.7.6 Read protected logic input with two address fields

The MSI 430-x/???- LEUZE module supports “Read SLC-protected logic input” according to the following table:

Tab. 8.22: Request data structure for Read SLC-protected logic input with two address fields

| Name | Data type | Description | Data range |
|---|---|---|---|
| Byte size | USINT | Number of data bytes to be read | Assembly instance 37: 0 to 50<br>Assembly instance 57: 0 to 67<br>Assembly instance 167: 0 to 112 |
| File number | USINT | Assembly instance ID | 37, 57, 167 |
| File type | USINT | Data type | 0x89 = Integer data |
| Element number | USINT | Offset = ID of the first element of the answer | Assembly instance 37: 0 to 24 – Size/2<br>Assembly instance 57: 0 to 33 – Size/2<br>Assembly instance 167: 0 to 55 – Size/2 |

Tab. 8.23: Feedback to the MSI 430-x/???- LEUZE module for Read SLC-protected logic input with two address fields

| Name | Data type | Description | Data range |
|---|---|---|---|
| Payload | UINT[n] | 2·n = Number of data bytes | 0 to 65535 |

#### 8.4.7.7 Write protected logic input with two address fields

The MSI 430-x/???- LEUZE module supports “Write SLC-protected logic input” according to the following table:

No support is required for assembly instances 138 to 141. The offset, i.e. the first byte, is instead specified by the **element number**.

Tab. 8.24: Request data structure for Write SLC-protected logic input with two address fields

| Name | Data type | Description | Data range |
|---|---|---|---|
| Byte size | USINT | Number of data bytes to be written | 0 to 50 |
| File number | USINT | Assembly instance ID | 37 |
| File type | USINT | Data type | 0x89 = Integer data |
| Element number | USINT | Offset = ID of the first element to be sent back | 0 to 24 – Size/2 |
| Payload | UINT[n] | n = Size/2 | 0 to 65535 |

#### 8.4.7.8 Read protected logic input with three address fields

The MSI 430-x/???- LEUZE module supports “Read SLC-protected logic input” according to the following table:

Tab. 8.25: Request data structure for Read SLC-protected logic input with three address fields

| Name | Data type | Description | Data range |
|---|---|---|---|
| Size | USINT | Number of data bytes to be read | Assembly instance 37: 0 to 50<br>Assembly instance 57: 0 to 67<br>Assembly instance 167: 0 to 112 |
| File number | USINT | Assembly instance ID | 37, 57, 167 |
| File type | USINT | Data type | 0x89 = Integer data |

| Name | Data type | Description | Data range |
|---|---|---|---|
| Element number | USINT | Offset = ID of the first element of the answer | Assembly instance 37: 0 to 24 – Size/2<br>Assembly instance 57: 0 to 33 – Size/2<br>Assembly instance 167: 0 to 55 – Size/2 |
| Subelement | USINT | Doesn't matter | 0 to 254 (for number of bytes 1) |

#### 8.4.7.9 Write protected logic input with three address fields

The MSI 430-x/???- LEUZE module supports "Write SLC-protected logic input" according to the following table:

Tab. 8.26: Request data structure for Write SLC-protected logic input with three address fields

| Name | Data type | Description | Data range |
|---|---|---|---|
| Size | USINT | Number of data bytes to be written | 0 to 50 |
| File number | USINT | Assembly instance ID | 37 |
| File type | USINT | Data type | 0x89 = Integer data |
| Element number | USINT | Offset = ID of the first element of the answer | 0 to 25 – Size/2 |
| Subelement | USINT | Doesn't matter | 0 to 254 (for number of bytes 1) |
| Payload | UINT[n] | n = Size/2 | 0 to 65535 |

### 8.4.8 Vendor object

The vendor object with class ID = 0x78 provides CRC, status and diagnostic data which are not covered by device profile **discrete universal I/O device**. Furthermore, it provides an interface to input and output data in a compressed format that therefore reduces network traffic.

Note that several instances have different attribute types and numbers. Several data items are packed together into this vendor object class for legacy reasons.

#### 8.4.8.1 Instance 1

Instance 1, attributes 1 to 50, supply input bytes configured by the **output data set 1** in MSI.designer. This is data transferred by the logic of the controller module to the PLC.

#### 8.4.8.2 Instance 2

Instance 2, attribute 1, supplies the CRC of the active project file created by MSI.designer. Instance 2, attributes 2 to 8 are reserved for future applications.

#### 8.4.8.3 Instance 3

Instance 3, attributes 1 to 60 make state bytes available. The descriptions for each bit are listed in the table *State bytes of the controller module MSI 420/430 [chapter 8.4.8.7]*. This data corresponds to **data set 3**, which is described at various points in this document.

A value = 1 for bits in instance 3, attributes 1 to 60, stands for "OK"/"Not used"/"Reserved". A value = 0 stands for "Fault" or "Error" or "Outside the limit". "Doesn't matter" means that the value can be equal to 0 or 1.

"EA module at Pos. n" with n = 1 .. 12 stands for the first to twelfth safe or unsafe expansion module.

#### 8.4.8.4 Instance 4

Instance 4, attributes 1 to 60 are reserved for future applications. Values are zero and changes are reserved.

#### 8.4.8.5 Instance 5

Attribute 1 of instance 5 provides the system status/mode of the controller module. The values are listed in the following table:

Tab. 8.27: System status/modes of the MSI 430-x/???- LEUZE module

| System state/mode | Value |
|---|---|
| Supply voltage A1/2 available | 0 |
| Initialization | 1 |
| Configuration / project file required | 2 |
| Configuration running / project file being downloaded | 3 |
| Idle | 4 |
| Run | 5 |
| Critical error | 7 |
| Force mode | 21 |

#### 8.4.8.6 Instance 6

Attribute 1 of instance 6 provides the error code of the most recent error of the controller module. A value = 0 means that no error has occurred. Attribute 2 supplies the error code of the previous error, etc., up to and including attribute 5.

#### 8.4.8.7 Instance 7

Attributes 1 to 50 of instance 7 represent the **input data blocks 1 to 5** in MSI.designer. They represent the data transferred by the PLC to the logic of the controller module.

Attributes 1 to 50 of instance 7 possess the same data as assembly instance 37, byte 0 to 49.

**NOTICE**

You can find an explanation of the technical terms used below here: *Abbreviations and definitions [chapter 1.5]*

Tab. 8.28: Overview of the vendor-specific object (Leuze electronic, 0x78), supported by the MSI 430-x/???- LEUZE module

| Class | Instance | Attribute | Access | Data type | Description | Data range |
|---|---|---|---|---|---|---|
| 120 | 0 = Class | 1 | Read | UINT | Revision of the class | 1 |
| 120 | 0 = Class | 2 | Read | UINT | Max. instance | 4 |
| 120 | 0 = Class | 3 | Read | UINT | Number of instances | 4 |
| 120 | 0 = Class | 5 | Read | UINT[3] | List of optional services | {2, 0x4c, 0x4d} |
| 120 | 0 = Class | 6 | Read | UINT | Max. class attribute ID | 7 |
| 120 | 0 = Class | 7 | Read | UINT | Max. instance attribute ID | 60 |
| 120 | 1 | n+1 | Read | USINT | The input byte “n”, configured by the **output data set 1** in MSI.designer, stands for the data transferred by the logic of the controller module to the PLC. The values n = 0 to 49 apply. | 0 to 255 |
| 120 | 2 | 1 | Read | UDINT | Project file CRC (data set 2) | 0 to $2^{32}$-1 |

| Class | Instance | Attribute | Access | Data type | Description | Data range |
|---|---|---|---|---|---|---|
| 120 | 2 | 2 to 8 | Read | UDINT | Reserved (data set 2) | 0 |
| 120 | 3 | n+1 | Read | BYTE | State byte "n" of the controller module, for which n = 0 to 59 | 0 to 255 |
| 120 | 4 | n+1 | Read | BYTE | Additional byte "n" of the controller module, for which n = 0 to 59 | 0 |
| 120 | 5 | 1 | Read | USINT | MSI 430-x/???- LEUZE system mode (*See [chapter 8.4.8.5]*) | 1, 2, 3, 4, 5, 7, 21 |
| 120 | 6 | n | Read | UDINT | Error code in the controller module, with n = 1 for the most recently occurred error, n = 2 for the previous error, etc., with n = 1 to 5 | 0 to $2^{32}$-1 |
| 120 | 6 | 1 | Write | UDINT | Clear error list in instance 6 | 0 |
| 120 | 7 | n+1 | Write, read | BYTE | Output bit "n", which is configured by the **input data blocks 1 to 5** in MSI.designer, stands for the data transferred by the PLC to the logic of the controller module. n = 0 to 49. | 0 to 255 |

Tab. 8.29: State bytes of the controller module MSI 420/430

| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **0** | Controller module State, voltage B2 | Controller module Collective error fast shut-off | Controller module State, voltage B1 | Controller module Configuration status | Controller module State, voltage A1/2 | Controller module External module state | Controller module Internal module state | Reserved |
| **1** | Controller module Output data state | Controller module Input data state | Reserved | Reserved | Controller module IQ3+IQ4 overcurrent | Controller module IQ1+IQ2 overcurrent | Controller module Q3+Q4 overcurrent | Controller module Q1+Q2 overcurrent |
| **2** | Controller module I8 Test pulse error | Controller module I7 Test pulse error | Controller module I6 Test pulse error | Controller module I5 Test pulse error | Controller module I4 Test pulse error | Controller module I3 Test pulse error | Controller module I2 Test pulse error | Controller module I1 Test pulse error |

| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **3** | Controller module I16 Test pulse error or HW limit frequency I16 | Controller module I15 Test pulse error or HW limit frequency I15 | Controller module I14 Test pulse error or HW limit frequency I14 | Controller module I13 Test pulse error or HW limit frequency I13 | Controller module I12 Test pulse error | Controller module I11 Test pulse error | Controller module I10 Test pulse error | Controller module I9 Test pulse error |
| **4** | Cable break at I16 | Cable break at I15 | Cable break at I14 | Cable break at I13 | Controller module IQ4 (input) Test pulse error | Controller module IQ3 (input) Test pulse error | Controller module IQ2 (input) Test pulse error | Controller module IQ1 (input) Test pulse error |
| **5** | Controller module I15/I16 Dual channel state | Controller module I13/I14 Dual channel state | Controller module I11/I12 Dual channel state | Controller module I9/ I10 Dual channel state | Controller module I7/ I8 Dual channel state | Controller module I5/ I6 Dual channel state | Controller module I3/ I4 Dual channel state | Controller module I1/ I2 Dual channel state |
| **6** | Inversion error I14 vs. I16 | Inversion error I13 vs. I15 | Frequency difference I14 vs. I16 | Frequency difference I13 vs. I15 | Phase difference I14 vs. I16 too low | Phase difference I13 vs. I15 too low | Controller module IQ3/IQ4 Dual channel state | Controller module IQ1/IQ2 Dual channel state |
| **7** | I16 Stuck-at low | I16 Stuck-at high | I15 Stuck-at low | I15 Stuck-at high | I14 Stuck-at low | I14 Stuck-at high | I13 Stuck-at low | I13 Stuck-at high |
| **8** | Controller module Q4 Stuck-at low | Controller module Q4 Stuck-at high | Controller module Q3 Stuck-at low | Controller module Q3 Stuck-at high | Controller module Q2 Stuck-at low | Controller module Q2 Stuck-at high | Controller module Q1 Stuck-at low | Controller module Q1 Stuck-at high |
| **9** | Controller module IQ4 (output) Stuck-at low | Controller module IQ4 (output) Stuck-at high | Controller module IQ3 (output) Stuck-at low | Controller module IQ3 (output) Stuck-at high | Controller module IQ2 (output) Stuck-at low | Controller module IQ2 (output) Stuck-at high | Controller module IQ1 (output) Stuck-at low | Controller module IQ1 (output) Stuck-at high |
| 10 | Reserved | 1. Gateway module Output data state | 1. Gateway module Input data state | 1. Gateway module Configuration state | Doesn’t matter | Reserved | 1. Gateway module Internal module state | Doesn’t matter |
| 11 | Reserved | 2. Gateway module Output data state | 2. Gateway module Input data state | 2. Gateway module Configu ration status | Doesn’t matter | Reserved | 2. Gateway module Internal module state | Doesn’t matter |

| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 8 + 4·n | Re-served | IO module at pos. n Collective error fast shut-off | IO module at pos. n State, voltage A1/2 (power supply for Q1 to Q4) | IO module at pos. n Configura-tion state | Doesn't matter | IO module at pos. n External module state | IO module at pos. n Internal module state | Doesn't matter |
| 9 + 4·n | IO mod-ule at pos. n Output data state | IO module at pos. n Input data state | Reserved | Reserved | IO module at pos. n I7/I8 Dual channel state | IO module at pos. n I5/I6 Dual channel state | IO module at pos. n I3/I4 Dual channel state | IO module at pos. n I2/I1 Dual channel state |
| 10 + 4·n | IO mod-ule at pos. n I8 test pulse er-ror | IO module at pos. n I7 test pulse er-ror | IO module at pos. n I6 test pulse er-ror | IO module at pos. n I5 test pulse er-ror | IO module at pos. n I4 test pulse er-ror | IO module at pos. n I3 test pulse er-ror | IO module at pos. n I2 test pulse er-ror | IO module at pos. n I1 test pulse er-ror |
| 11 + 4·n | IO mod-ule at pos. n Q4 Stuck-at low | IO module at pos. n Q4 Stuck-at high | IO module at pos. n Q3 Stuck-at low | IO module at pos. n Q3 Stuck-at high | IO module at pos. n Q2 Stuck-at low | IO module at pos. n Q2 Stuck-at high | IO module at pos. n Q1 Stuck-at low | IO module at pos. n Q1 Stuck-at high |

## 8.5 Supported assembly data

Assemblies are collections of data attributes and are optimized for high performance and a low telegram overhead. The MSI 430-x/???- LEUZE module supports a series of predefined, static assembly instances for input and output data. Access is possible via various instances of the CIP assembly object. In addition, access is possible both via the implicit and explicit message transmission. The assembly size is variable. It is thus possible to request parts of an assembly. The following table (*Overview of assembly data bytes of the MSI 430-x/???- LEUZE module [chapter 8.5.1]*) offers an overview of the supported assembly instances and the meaning of the transmitted data.

### 8.5.1 List of assembly data

Tab. 8.30: Overview of assembly data bytes of the MSI 430-x/???- LEUZE module

| Instance | Byte | Access | Data type | Description | Size | Data range |
|---|---|---|---|---|---|---|
| 37 | 0 to 49 | Write, read | BYTE[50] | Logic output bytes, configura-tion via **Input data block 1 to 5** in MSI.designer (*see [chapter 8.5.2]*) | 1 to 50 Bytes | 0 to 0xff |
| 138 | 10 to 49 | Write, read | BYTE[40] | Logic output bytes, configura-tion via **Input data block 2 to 5** in MSI.designer (*see [chapter 8.5.2]*) | 1 to 40 Bytes | 0 to 0xff |
| 139 | 20 to 49 | Write, read | BYTE[30] | Logic output bytes, configura-tion via **Input data block 3 to 5** in MSI.designer (*see [chapter 8.5.2]*) | 1 to 30 Bytes | 0 to 0xff |
| 140 | 30 to 49 | Write, read | BYTE[20] | Logic output bytes, configura-tion via **Input data block 4 and 5** in MSI.designer (*see [chapter 8.5.2]*) | 1 to 20 Bytes | 0 to 0xff |

| Instance | Byte | Access | Data type | Description | Size | Data range |
|---|---|---|---|---|---|---|
| 141 | 40 to 49 | Write, read | BYTE[10] | Logic output bytes, configuration via **Input data block 5** in MSI.designer (*see [chapter 8.5.2]*) | 1 to 10 Bytes | 0 to 0xff |
| 57 | 0 to 49 | Read | BYTE[50] | Logic output bytes, configuration via **Output data block 1** in MSI.designer (*see [chapter 8.5.3]*) | 1 to 50 Bytes | 0 to 0xff |
| | 50 to 65 | Read | BYTE[16] | Values of the input terminals Ix | 1 to 16 Bytes | 0 to 0xff |
| | 66 | Read | BYTE | Input and output state | 1 bytes | 0x00, 0x40, 0x80, 0xc0 |
| 167 | 0 to 49 | Read | BYTE[50] | Logic output bytes, configuration via **Output data block 1** in MSI.designer (*see [chapter 8.5.3]*) | 1 to 50 Bytes | 0 to 0xff |
| | 50 | Read | BYTE | Bit 7: Input state<br>Bit 6: Output state<br>Bit 5: Error code ≠ 0 | 1 bytes | 0x00, 0x40, 0x80, 0xc0 |
| | 51 | Read | BYTE | System mode | 1 bytes | 1, 2, 3, 4, 5, 7, 21 |
| | 52 to 111 | Read | BYTE[60] | Status bytes of the controller module (*Instance 3 of class 120 [chapter 8.4.8.3]*), **output data set 3** (*see [chapter 8.5.3]*) | 60 bytes | 0 to 0xff |

The data type of supported assemblies is BYTE, which means strings of 8 bits each. The naming in Logix Designer is SINT, which has the same size of 8 bits each.

If the PLC requires a configuration assembly, any value or even no value can be used for the **assembly instance**. The **size** of the configuration assembly must be zero.

The assembly instances for **Input** and **Output** are listed in Table "*Overview of assembly data bytes from the module [chapter 8.5.1]*". These settings can be used in generic EtherNet module configuration in Logix Designer (see illustration below).

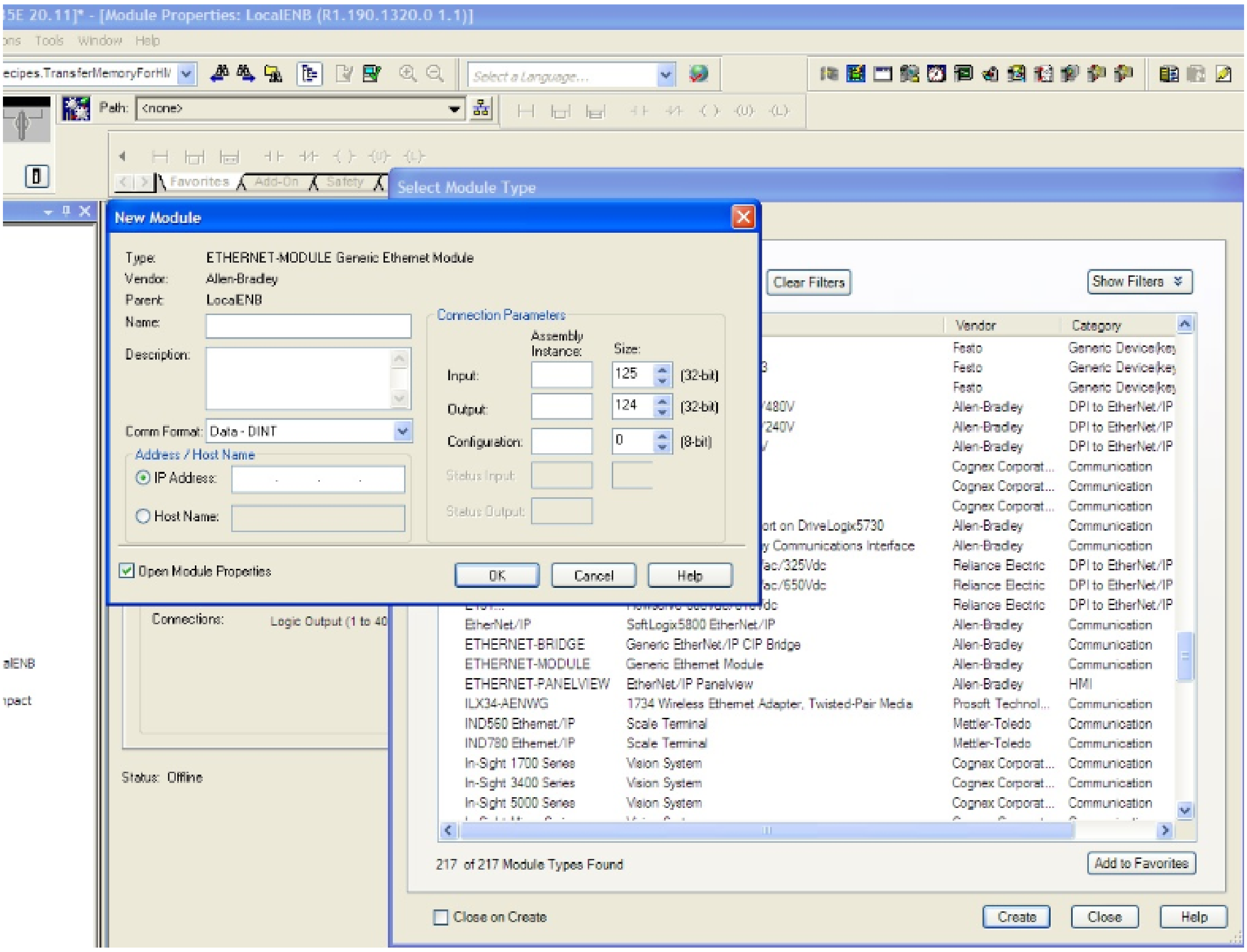



Fig. 8.13: Generic EtherNet module configuration

## 8.5.2 Assembly instances for logic output bytes

### 8.5.2.1 Assembly instance 37 = 0x25

Assembly instance 37 belongs to the device profile **discrete universal I/O device.** It contains output data (O→T) with a scope of up to 50 bytes.

Assembly instance 37 corresponds to **input data block 1 to 5** of the logic data of the MSI 430-x/???-LEUZE module with a total scope of 50 bytes.

### 8.5.2.2 Assembly instances 138 = 0x8a to 141 = 0x8d

Assembly instances 138 to 141 are provided to make more than one output data connection available. In Class 1 connections, output data from the PLC to the controller module can only be sent using “Exclusive Owner” rights. If, for example, a PLC “possesses” the assembly instance 138, then it “possesses” the output bytes 10 to 49. By contrast, output bytes 0 to 9 are freely available and can be used by another PLC (O→T).

In a further example, the first PLC “possesses” 10 output bytes of the assembly instance 37, whilst the second PLC “possesses” 10 output bytes of assembly instance 138 and the third assembly instance 139 with 30 output bytes. Here, three PLCs possess “Exclusive Owner” connections with output data. In total, up to five PLCs can share the output data range, each with 10 bytes.

Assembly instance 138 comprises data with a scope of up to 40 bytes, assembly instance 139 comprises data with a scope of up to 30 bytes, assembly instance 140 comprises data with a scope of up to 20 bytes and assembly instance 141 comprises data with a scope of up to 10 bytes.

The first byte of assembly instance 138 is the eleventh byte of the logic data of the MSI 430-x/???- LEUZE module. In MSI.designer, it has the designation **input data block 2**. The first byte of assembly instance 139 is 21. Byte of the logic data of module MSI 430-x/???- LEUZE. In MSI.designer, it has the designation **input data block 3**. The first byte of assembly instance 140 is 31. Byte of the logic data of module MSI

430-x/???- LEUZE. In MSI.designer, it has the designation **input data block 4**. The first byte of assembly instance 141 is 41. Byte of the logic data of module MSI 430-x/???- LEUZE. In MSI.designer, it has the designation **input data block 5**.

Write requests are refused if the assembly is already used by an active I/O connection.

## 8.5.3 Assembly instances for logic input bytes

### 8.5.3.1 Assembly instance 57 = 0x39

Assembly instance 57 belongs to the device profile **discrete universal I/O device.** It contains output data (T→O) with a scope of up to 67 bytes.

The first 50 bytes of assembly instance 57 correspond to the **output data set 1** of the logic data of the MSI 430-x/???- LEUZE module. The following table explains the meaning of bytes 50 to 66: *Data of the class 1 connection "Logic output (1 to 400) and logic/physical input" [chapter 8.3.1]*

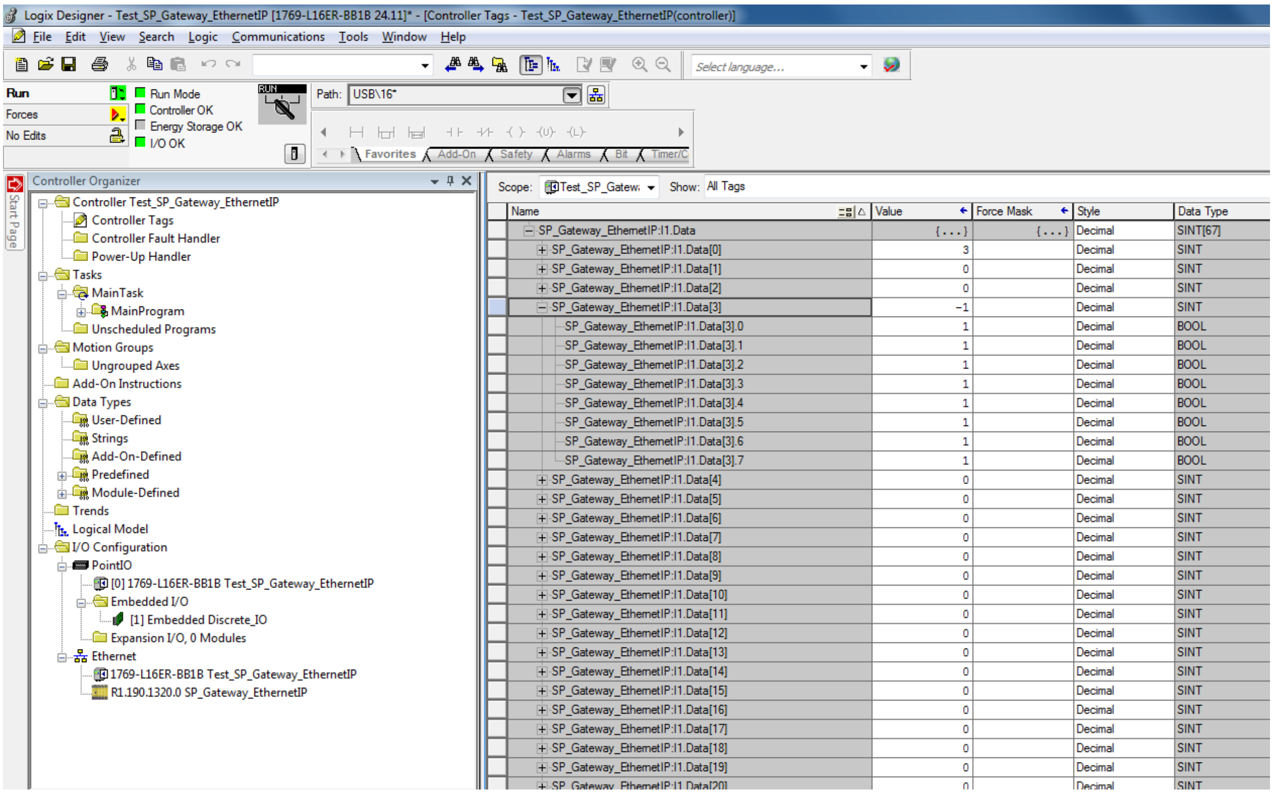



Fig. 8.14: Example of the display of assembly instance 57 in the Logix Designer

### 8.5.3.2 Assembly instance 167 = 0xa7

Assembly instance 167 possesses a different data structure to instance 57. Instance 167 makes the data available in the MSI 400 system in more detail.

Assembly instance 167 in the MSI 400 system comprises data (T→O) with a scope of up to 112 bytes.

#### 8.5.3.2.1 Bytes 0 to 49

Assembly instance 167 corresponds to **output data set 1** of the logic data of the MSI 430-x/???- LEUZE module with a total scope of 50 bytes.

Here, attributes 1 to 50 are represented as with instance 57.

#### 8.5.3.2.2 Byte 50

Bit 7 of byte 50 of assembly instance 167 has the same value as class 29 instance 1 attribute 5, which represents the group status of all input terminals.

Bit 6 of byte 50 of assembly instance 167 has the same value as class 30 instance 1 attribute 5, which represents the group status of all output terminals.

Bit 5 of byte 50 of assembly instance 167 indicates that an error code is pending in class 120 instance 6 attribute 1.

Bits 0 to 4 of byte 50 of assembly instance 167 are reserved for future use.

8.5.3.2.3 Byte 51

Byte 51 of assembly instance 167 supplies the system mode of the controller modules. It shows the same value as attribute 1 of instance 5 in class 120.

8.5.3.2.4 Bytes 52 to 111

Bytes 52 to 111 of assembly instance 167 make the corresponding state bytes of the controller mode available. They show the same value as attributes 1 to 60 of instance 3 in class 120.

## 8.6 Access to CIP objects

### 8.6.1 Explicit messaging

Explicit message transmission uses the TCP/IP protocol as well as an EtherNet/IP-specific encapsulation layer. Explicit message transmission can be connection-free (UCMM) and connected, e.g. session-based. The latter is termed **Class 3 Messaging**. Both UCMM and Class 3 use an EPATH to address the required data. An EPATH is made up of the service, class, instance and attribute ID.

With explicit message transmission, each attribute of the following objects can be accessed:

- *Identity class (0x01) [chapter 8.4.1]*
- *Assembly class (0x04) [chapter 8.4.2]*
- *Discrete input point object (0x08) [chapter 8.4.3]*
- *Discrete output point object [chapter 8.4.4]*
- *Discrete input group object (0x1D) [chapter 8.4.5]*
- *Discrete output group object (0x1D) [chapter 8.4.6]*
- *Vendor-specific object (0x78) [chapter 8.4.8.7]*

Each request must possess a valid EPATH referring to the required object/attribute. The appropriate attribute can be read using the GetAttributeSingle service, if it is labeled as **Read** in these tables. The appropriate attribute can be written using the SetAttributeSingle service, if it is labeled as **Write** in these tables.

### 8.6.2 Implicit messaging

Implicit message transmission uses EtherNet/IP, the UDP/IP protocol as well as an EtherNet/IP-specific encapsulation layer. Implicit message transmission is also termed **Transport Class 1**. The PLC can set up a Class 1 connection with the MSI 430-x/???- LEUZE module, by placing the service request **Forward_Open** with it. This configures connection information for exchanging input/output data, e.g. the RPI unicast or multicast connections, amongst others. Class 1 connections only support assemblies for the exchange of input/output data or “wild cards” to signal data-free heartbeat connections. Configuration assemblies are accepted as part of the Forward_Open-Service, with the exception of TCP/IP objects (Class 0xF5), although they are not processed by the MSI 430-x/???- LEUZE module.

As the configuration details of the connection are only sent once in the Forward_Open-Frame, implicit message transmission is aligned to performance and has a lower telegram overhead than explicit message transmission. Assembly instances possess predefined attributes in a specific order. Nonetheless, the sender, i.e. the PLC; specifies the data size in Forward_Open during the setup of the Class 1 connection. This means that only data byte from the beginning of the instance up to the specified size are exchanged.

The MSI 430-x/???- LEUZE module supports seven static assembly instances. These are listed in the table *Overview of the assembly class (0x04) supported by the MSI 430-x/???- LEUZE module [chapter 8.4.2]*. All data members of the instance has fixed coding. Dynamic assembly instances are not currently supported by the MSI 430-x/???- LEUZE module.

An I/O assembly contains either input or output data, but not both at the same time. The following figure shows the data flow when multiple assembly instances are used. Predefined assemblies are interconnected by blue lines, vendor-specific assemblies by black lines. The controller module is shown as a hatched rectangle.

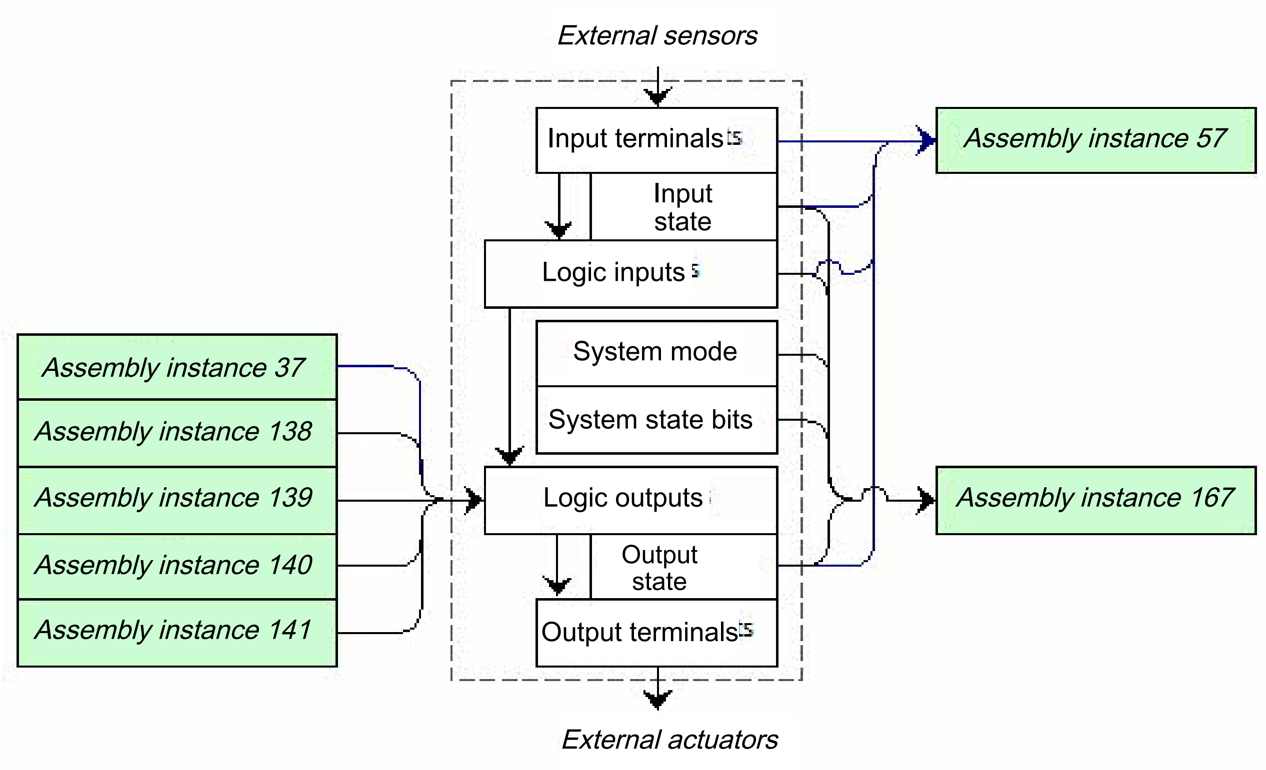



Fig. 8.15: Data flow when using assembly instances of the MSI 430-x/???- LEUZE module.

### 8.6.3 Symbolic addressing

In addition to the addressing of assembly instances, symbolic addressing by name is also possible by selecting connections.

In MSI.designer, tag names can be changed in the **Gateway configuration** tab.

**Device functionality**

Output data (to PLC):

| Dataset | Byte | Tag/File-Name | Assembly Instance | Byte |
|---|---|---|---|---|
| 1 | 0..49 | OutDataSet1 | 57 | 0..66 |
| 2 | 0..31 | OutDataSet2 | - | - |
| 3 | 0..59 | OutDataSet3 | 167 | 0..111 |
| 4 | 0..59 | OutDataSet4 | - | - |

Input data (from PLC):

| Data block | Byte | Tag/File-Name | Assembly Instance | Byte |
|---|---|---|---|---|
| 1 | 0..9 | InDataSet1 | 37 | 0..49 |
| 2 | 10..19 | InDataSet2 | 138 | 10..49 |
| 3 | 20..29 | InDataSet3 | 139 | 20..49 |
| 4 | 30..39 | InDataSet4 | 140 | 30..49 |
| 5 | 40..49 | InDataSet5 | 141 | 40..49 |

Fig. 8.16: Configuration of symbolic names for assemblies in MSI.designer

**NOTICE**

The functions of the UCMM Message Client (unconnected), which can also be configured in MSI.designer, are not available in the module version D-01.01.

## 8.7 Optimizing performance

A configuration of the number of process data bytes exactly matching the application helps to reduce the volume of periodically exchanged data bytes.

The PLC specifies the number of output bytes in the form of specific Forward_Open Service data as **Connection Size** for O→T. The PLC should set the **Fixed/Variable** bit to 1, meaning **variable**.

The PLC also specifies the number of input bytes. The controller module cyclically transfers data in the scope of the **connection size** for T→O through RPI in the value specified in Forward_Service. If the **fixed/ variable** bit is set by the PLC, meaning **variable**, then not all the assembly bytes must be transmitted.

## 8.8 Connection with more than one PLC

The EtherNet/IP function of the MSI 430-x/???- LEUZE module allows access by more than one PLC. Up to five encapsulation sessions with input and output data can be set up simultaneously.

If only reading the process data of the MSI 430-x/???- LEUZE module is required, “Input only” or “Listen only” connections can also be used. Note that a “Listen only” connection is closed automatically by the MSI 430-x/???- LEUZE module when the owner, who has set up the “Exclusive” or “Input Only” connection, terminates the connection.

If process data from multiple PLCs are to be transmitted to the MSI 430-x/???- LEUZE module, the other PLCs can access the assembly instances 138 to 141 for Class 1 connections. Class 3 connections can be set up in parallel, provided that there is no conflict with regard to the owner. Please see the following for more information: *List of assembly data [chapter 8.5.1]*

## 8.9 Troubleshooting and eliminating errors

### 8.9.1 Notifications via the network

#### 8.9.1.1 Explicit messaging connection

The device status is available by reading class 1, instance 1, attribute 5. Vendor-specific interface for alarms and diagnostic functions for explicit message connections is defined as follows:

The presence of an alarm can be checked by reading class 29, instance 1, attribute 5 and class 30, instance 1, attribute 5. The module mode (**Run** or another state) has to be checked, because the alarm bit is always set to 0 = OK every time the module is not in **Run** mode.

The module mode can be detected by reading class 120 instance 5 attribute 1.

The presence of diagnostic events can be checked by reading class 120 instance 6 attribute 1.

Detailed reasons for process alarms and system diagnostic events can be identified by reading all 60 attributes of class 210 instance 3, which contains the dedicated system status bytes.

#### 8.9.1.2 Implicit messaging connection

If assembly instance 57 is used, bit 6 and 7 of byte 66 signal a process alarm.

If assembly instance 167 is used, bit 6 and 7 of byte 66 signal a process alarm. Bit 5 signals diagnostic events or process alarms when set.

Event details can be queried through explicit message requests, as described here: *Explicit messaging [chapter 8.6.1]*

### 8.9.2 LED states

#### 8.9.2.1 MS (module status)

The MSI 430-x/???- LEUZE module possesses a two-color (red/green) LED with the designation **MS**. This is the **Module Status Indicator**.

The Module Status Indicator is *dark*, if no power supply is connected. It *flashes green* if the device has not been configured. It *turns green* if the device is running correctly. It *flashes green/red* if the device is performing a switch-on test.

The Module Status Indicator *flashes red* if EtherNet/IP is activated and the device has detected a serious, eliminable error. A faulty project file or one which does not match the hardware is classified as a serious, eliminable error. The display *turns red* if EtherNet/IP is activated and the device has detected a serious, non-eliminable error and there is a **Critical Fault**.

Tab. 8.31: MS-LED state (Selection)

| Project file | System mode | Ext. Error | MS-LED state |
|---|---|---|---|
| Doesn't matter | Switching on | doesn't matter | Green -> Red |
| Deleted | Init | Doesn't matter | Flashing green |
| Invalid | Init | Doesn't matter | Flashing red |
| Valid | Idle mode | Doesn't matter | Flashing green |
| Valid | Run | No | Turns green |
| Valid | Run | Yes | Turns green/red or flashes red |
| Valid | Critical error | Doesn't matter | Turns red |

#### 8.9.2.2 NET (network status)

The MSI 430-x/???- LEUZE module possesses a two-color (red/green) LED with the designation **NET**. This is the **Network Status Indicator**.

Tab. 8.32: Meaning of the NET LED (used as EtherNet/IP gateway)

| NET LED | Meaning / reason |
|---|---|
| LED off | • Power supply not connected.<br>or<br>• Power supply connected but IP address not configured. |
| Green (1 Hz) | EtherNet/IP has been activated and an IP address has been configured but there is no CIP connection and an "Exclusive Owner" connection shows no time-out. |
| Green | An IP address has been configured, there is at least one CIP connection (of any transport class) and an "Exclusive Owner" connection has not yet shown a time-out. |
| / Red/green | During power-on test |
| Red | EtherNet/IP has been activated, an IP address has been configured and an "Exclusive Owner" connection, for which the device is the target device, has shown a time-out. |

The Network Status Indicator is *dark* if no power supply is connected or a power supply is connected but no IP address is configured (interface configuration attribute of the TCP/IP interface object). It *flashes green* if EtherNet/IP is activated and an IP address has been configured but no CIP connection is available and an "Exclusive Owner" connection has not yet shown a time-out. It *turns green* if an IP address has been configured, there is at least one CIP connection (of any transport class) and an "Exclusive Owner" connection has not yet shown a time-out. It *flashes green/red* if the device is performing a switch-on test.

The Network Status Indicator *flashes red* if EtherNet/IP is activated, an IP address has been configured and an "Exclusive Owner" connection, for which the device is the target device, has shown a time-out. The Network Status Indicator only turns green again when all the expired "Exclusive Owner" connections have been restored. The Network Status Indicator switches from flashing red to being lit in green if all the connections of the previously expired O->T connection points have been restored. Time-outs in other connections than the "Exclusive Owner" connections do not result in the indicator flashing red. The "flashing red" state only applies to connections with the target device. PLCs and CIP routers do not instigate a transition to this state if a created or routed connection shows a time-out.

Tab. 8.33: Troubleshooting on the MSI 430-x/???- LEUZE module (use as EtherNet/IP gateway)

| Error | | Possible cause | Possible remedy |
|---|---|---|---|
| Key: ○ LED off / ✷ LED flashes / ● LED lights up | | | |
| The MSI 430-x/???- LEUZE module does not provide any data. | | • The MSI 430-x/???- LEUZE module has been configured for data transmission to the PLC, but no Ethernet communication has been established or it is faulty.<br>• Duplicate IP address detected. Another network device has the same IP address.<br>• Incorrectly formatted PROFINET device name | • PROFINET IO must be activated in the project file. At least one Ethernet link must be established. Check the Ethernet wiring, check the Ethernet settings in the PLC and in MSI.designer.<br>• Correct the IP address and switch the system off and on again.<br>• Compare the device name of the PROFINET master and the MSI 430-x/???- LEUZE module. |
| LED PWR/ EC | ● Green | | |
| LED LINK | ● Green | | |
| LED ACT | ✷ Yellow | | |
| MS LED | ● Green | | |
| The MSI 430-x/???- LEUZE module does not provide any data. | | • Configuration required.<br>• The configuration has not yet been fully transmitted.<br>• The module version does not support any PROFINET IO. | • Configure the MSI 430-x/???- LEUZE module with a project file in which PROFINET IO is activated, and transfer the configuration to the MSI 430-x/???- LEUZE module.<br>• Use a device starting with module MSI 430-x/???- LEUZE version B-xx. |
| LED PWR/ EC | ● Green | | |
| LED LINK | ● Green | | |
| LED ACT | ✷ Yellow | | |
| MS LED | ✷ / ✷ Red/green | | |
| The MSI 430-x/???- LEUZE module does not provide any data. | | • The MSI 400 system is in the stop state. | • Start the controller module (switch to Run mode). |
| LED PWR | ● Green | | |
| LED LINK | ● Green | | |
| LED ACT | ✷ Yellow | | |
| MS LED | ✷ Green (1 Hz) | | |
| The MSI 430-x/???- LEUZE module does not provide any data. | | • The IP address for the MSI 430-x/???- LEUZE module is assigned by a DHCP server. Following a restart of the MSI 430-x/???- LEUZE module or the DHCP server, another address was allocated to the MSI 430-x/???- LEUZE module, which is unknown to the PLC. | • Either assign a permanent IP address to the MSI 430-x/???- LEUZE module or reserve a permanent IP address for the MSI 430-x/???- LEUZE module in the DHCP server (manual assignment using the MAC address of the MSI 430-x/???- LEUZE module). |
| LED PWR/ EC | ● Green | | |
| LED LINK | ● Green | | |
| LED ACT | ✷ Yellow | | |
| MS LED | ● Green | | |

| Error | Possible cause | Possible remedy |
|---|---|---|
| The MSI 430-x/???-LEUZE module / the MSI 400 system is in a Critical Error status. | • The MSI 430-x/???-LEUZE module is not properly connected to the other MSI 400 modules.<br>• The module connection plug is dirty or damaged.<br>• Another MSI 400 module has an internal critical error. | • Insert the I/O module correctly. Clean the connection plug and socket.<br>• Switch on the power supply again.<br>• Check the other MSI 400 modules. |

#### 8.9.2.3 LINK

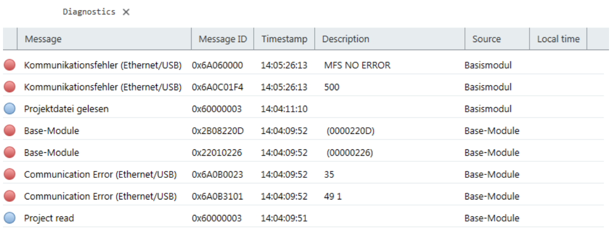
The MSI 430-x/???- LEUZE module possesses a green LED with the designation **LINK**. If there is no Ethernet connection, it stays dark. If there is a connection, it switches on.

#### 8.9.2.4 ACT (activity status)

The MSI 430-x/???- LEUZE module possesses a green LED with the designation **ACT**. If no port activity can be detected, it stays dark. If port activity is detected, it switches on.

### 8.9.3 Diagnostic functions in the configuration software

Additional diagnostic functions are available on the SD card using a log file with the name history.csv. In addition, the last entries are available in MSI.designer in the **Diagnostics** view. The timestamp in the **Local time** column provides information about how long the device has been switched on in total.




**Synchronize time**

With the safety controller connected you can synchronize the time on the safety controller with the time on the connected diagnostics computer. Even if you disconnect the connection to the controller, the **Diagnostics** view remains active, as long as the associated MSI.designer project is open.

| | NOTICE |
|---|---|
| ! | **Instructions in software manual**<br>You can find step-by-step instructions on how to synchronize the time here:<br>Software manual, chapter "Synchronize time for diagnostic purposes" |

## 8.10 Status bits

The EtherNet/IP gateway MSI-EN-IP sets status bits, which are available in the logic editor of MSI.designer for processing.

Tab. 8.34: Meaning of the state bits MSI-EN-IP[0] in the logic editor

| Name of the state bits | Set to 1, if ... | Reset to 0 ... |
|---|---|---|
| Output status | ... a GetAttribute command was processed successfully,<br>or<br>... data of transport class 1 were sent to a PLC without errors. | ... if a connection of transport class 1 (implicit connection) was terminated and no further connection exists. |
| Input status | ... a SetAttribute command was processed successfully,<br>or<br>... data of transport class 1 received without error (consumed), whereby heartbeat data of connection point 198 from the PLC does not count | ... a connection of transport class 1 (implicit connection) was terminated for one of the connection points 57, 138, 139, 140 or 141 and no further connection exists to these connection points. |
| Internal state | ... the EtherNet/IP function of the module is ready for communication. | ... the EtherNet/IP function of the module is not ready for communication. |

# 9 PROFIBUS DP gateway

The following MSI 400 gateway can be used for PROFIBUS DP:

- MSI-FB-PROFIBUS

## 9.1 Interfaces and operation

### Operating and display elements

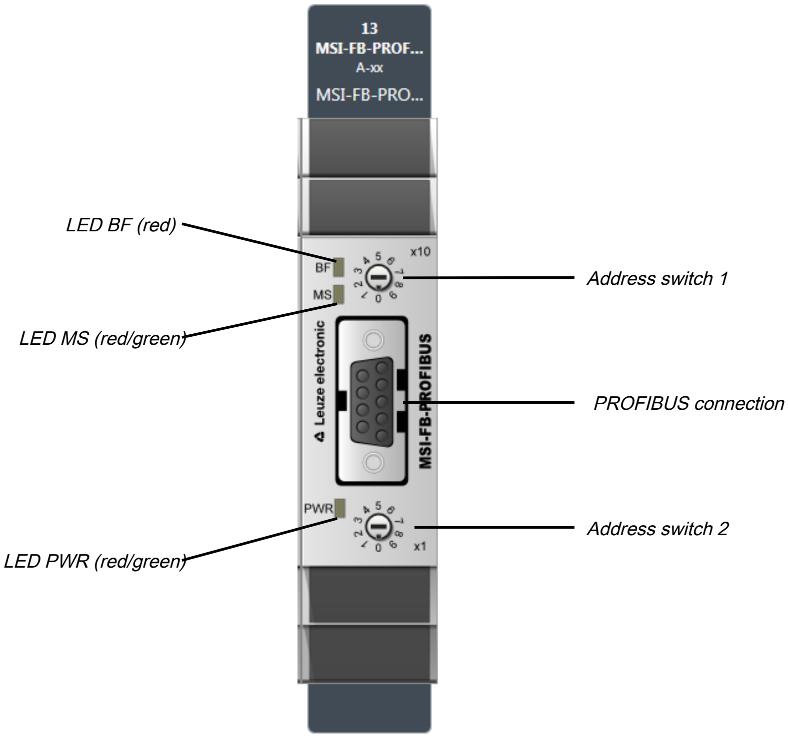



Fig. 9.1: Operating and display elements of the MSI-FB-PROFIBUS module

Tab. 9.1: Meaning of the state LEDs on the MSI-FB-PROFIBUS module

| LED | | Meaning |
|---|---|---|
| Key: ○ LED off / ☀ LED flashes / ● LED lights up | | |
| BF | ○ Off | Connection to the DP master established |
| | ● Red | No bus connection: Field bus cabling interrupted, address error or the master is no longer transmitting to the bus |
| MS | ○ Off | Power supply switched on, waiting for bus-off |
| | ● Green | Run |
| | ● Green | Stop |
| | ☀ / ☀ Red / green | Run, but the gateway has a fault |
| | ☀ Red | 1 Hz: Configuration required or is taking place right now<br>2 Hz: Critical error on the gateway |
| | ● Red | Critical error on another module |
| PWR | ○ Off | No power supply |
| | ● Green | Power supply switched on, no error |
| | ● Red | Critical error |

Tab. 9.2: Address switch of the MSI-FB-PROFIBUS module

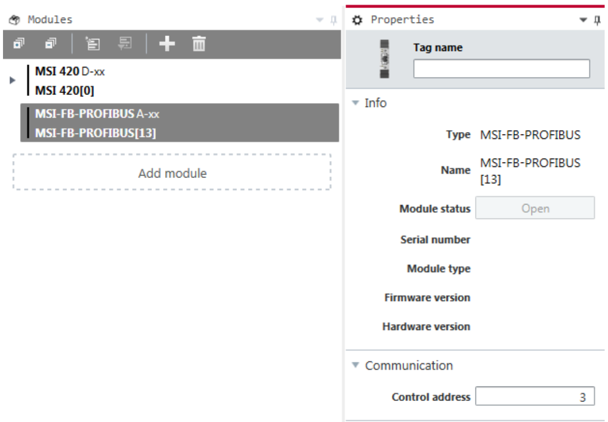
| Switches | Function |
|---|---|
| × 10 | Address switch 1<br>Rotary switch with 10 positions for setting the module address<br>(in tens) |
| × 1 | Address switch 2<br>Rotary switch with 10 positions for setting the module address<br>(in units) |

**How to set the PROFIBUS-DP address with the aid of the hardware address switches:**

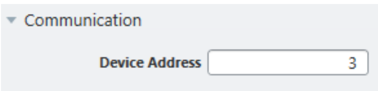
- ↳ Use the hardware address switches at the front of the system to set the PROFIBUS-DP address.
- ↳ Switch the MSI 400 system off and on again.

**How to set the PROFIBUS-DP address in the software:**

- ↳ Set the two hardware address switches on the front of the device to "00".
- ↳ Launch MSI.designer.
- ↳ Read in the hardware configuration, including the PROFIBUS-DP gateway. Instructions: Software manual, chapter "Connecting to the safety controller"
- ↳ Open the **Modules** docking window and select the MSI-FB-PROFIBUS module.
- ↳ Also open the **Properties** docking window.
  - ⇨ You see the configuration dialog for the MSI-FB-PROFIBUS module.




↳ Enter the desired value for the **Control address** parameter under **Communication**.




| NOTICE |
|---|
| ↳ You can set an address within the 1 ... 99 range with the aid of the hardware address switches. |
| ↳ You can set an address within the 3 ... 125 range with the aid of the MSI.designer. |
| ↳ The PROFIBUS master cannot overwrite the address. |
| ↳ An amended address setting will only become effective once you have switched off the MSI 400 system and switched it on again. |
| ↳ In the online mode, you can read out the address set at the PROFIBUS-DP gateway by clicking on the **Read** button above the **PROFIBUS address** field. |

**Pin assignment**

Connection to the PROFIBUS-DP field bus is via a 9-pin D-sub socket.

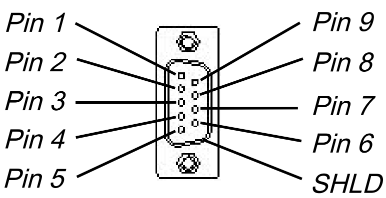



Fig. 9.2: Pin configuration of D-sub socket and plug for the MSI-FB-PROFIBUS module

Tab. 9.3: Reference for pin configuration

| Pin | Description |
|---|---|
| 1 | NC |
| 2 | NC |
| 3 | RxD/TxD-P |
| 4 | CNTR-P |
| 5 | GND-EXT |
| 6 | +5V-EXT |
| 7 | NC |
| 8 | RxD/TxD-N |
| 9 | CNTR-N (GND-EXT) |
| SHLD | Screening |

### Bus cable

The bus topology for PROFIBUS DP is a linear structure consisting of a screened and twisted 2-lead cable with active bus termination at both ends. The potential bus lengths range from 100 m at 12 kbit/s to 1200 m at 94 kbit/s.

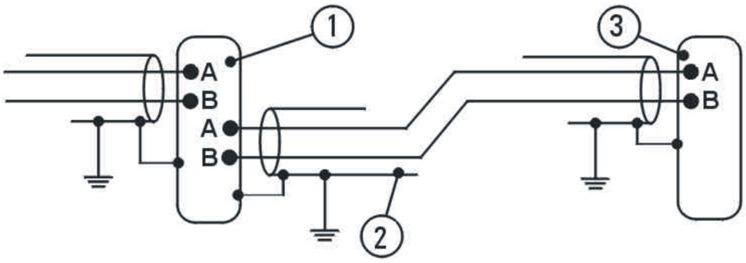



Fig. 9.3: Bus cable of the MSI-FB-PROFIBUS module

Tab. 9.4: Reference for pin configuration

| Position | Description |
|---|---|
| 1 | PROFIBUS user gray |
| 2 | Screened bus cable |
| 3 | PROFIBUS termination yellow (with integrated terminal resistances) |

### Line parameters

The bus cable characteristics have been defined in EN 50170 as cable type A.

Tab. 9.5: Line parameters of the MSI-FB-PROFIBUS module

| Characteristic | Value |
|---|---|
| Wave resistance | 135-165 Ω (at a frequency of 3-20 MHz) |
| Capacity per length unit | < 30 pF/m |
| Loop resistance | ≤ 110 Ω/km |
| Lead diameter | > 0.64 mm |
| Wire cross-section | > 0.34 $mm^2$ |

These cable parameters provide the following maximum physical dimensions for a bus section:

Tab. 9.6: Maximum line lengths of the MSI-FB-PROFIBUS module

| Baud rate (kbit/s) | Maximum cable length (m) |
|---|---|
| 9.6 | 1200 |
| 19.2 | 1200 |
| 93.75 | 1200 |
| 187.5 | 1000 |
| 500 | 400 |
| 1500 | 200 |
| 12000 | 100 |

### Data transmission rate

The data transmission rate is automatically set. The maximum baud rate is 12 Mbit/s.

## 9.2 Projecting

### GSD file

Under normal circumstances, the MSI-FB-PROFIBUS module is operated on a DP master that reads the device characteristics from the GSD file.

You will find the GSDML file and the equipment symbol for integration in a PLC of the product website of the MSI-FB-PROFIBUS on the Internet (www.leuze-shop.com).

### Process data transmitted by the MSI-FB-PROFIBUS module

The GSD file of the MSI-FB-PROFIBUS module provides input and output data blocks (virtual I/O device modules), which contain the process data. These 5 blocks must be projected in a natural sequence (1, 2, 3, 4, 5) in a DP configurator. No other sequence is possible.

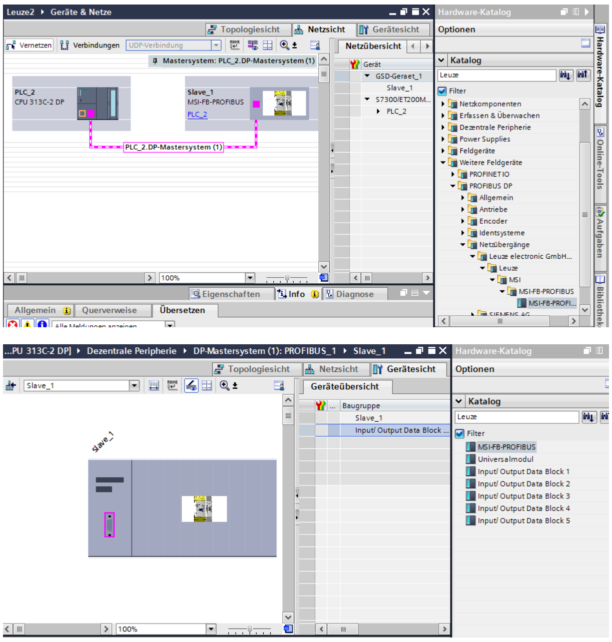



Fig. 9.4: Example for a PROFIBUS-DP configuration in the Siemens SIMATIC manager

**NOTICE**

- Depending on the PLC used, further modules may be shown (e.g. "Universal module"). These modules are not required and should be ignored.
- Data blocks 1–4 each contain 12 bytes, while data block 5 contains 2 bytes.

The content of the data blocks can be freely selected, but has been preconfigured as follows in the MSI.designer:

Tab. 9.7: Predefined content of input data block 1–5 of the MSI-FB-PROFIBUS module

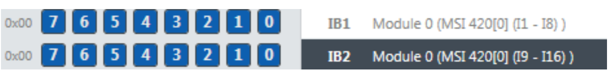
| | Data block 1 | Data block 2 | Data block 3 | Data block 4 | Data block 5 |
|---|---|---|---|---|---|
| | Output data block | Output data block | Output data block | Output data block | Output data block |
| **Byte 0** | Input values for Module 0 (I1..I8) | Input values for Module 1 | Output values for Module 1 | Not allocated | Not allocated |
| **Byte 1** | Input values for Module 0 (I9..I16) | Input values for Module 2 | Output values for Module 2 | Not allocated | Not allocated |
| **Byte 2** | Input values for Module 0 (IQ1..IQ4) | Input values for Module 3 | Output values for Module 3 | Not allocated | Not available |
| **Byte 3** | Output values for Module 0 (Q1..Q4,IQ1-IQ4) | Input values for Module 4 | Output values for Module 4 | Not allocated | |
| **Byte 4** | Direct data (Off) 0 | Input values for Module 5 | Output values for Module 5 | Not allocated | |
| **Byte 5** | Direct data (Off) 1 | Input values for Module 6 | Output values for Module 6 | Not allocated | |
| **Byte 6** | Direct data (Off) 2 | Input values for Module 7 | Output values for Module 7 | Not allocated | |
| **Byte 7** | Direct data (Off) 3 | Input values for Module 8 | Output values for Module 8 | Not allocated | |
| **Byte 8** | Direct data (Off) 4 | Input values for Module 9 | Output values for Module 9 | Not allocated | |
| **Byte 9** | Direct data (Off) 5 | Input values for Module 10 | Output values for Module 10 | Not allocated | |
| **Byte 10** | Direct data (Off) 6 | Input values for Module 11 | Output values for Module 11 | Not allocated | |
| **Byte 11** | Direct data (Off) 7 | Input values for Module 12 | Output values for Module 12 | Not allocated | |
| **Length** | 12 bytes | 12 bytes | 12 bytes | 12 bytes | 2 bytes |
| **Start address** | 1 | 13 | 25 | 37 | 49 |

Detailed information about the content of the process diagram may be found here: *Data transferred to the network (network input data sets [chapter 3.2]*).

**Delete any bytes not required**

You can delete bytes pre-allocated by MSI.designer that you do not require by clicking on them with the mouse.

- Launch MSI.designer.
- Read in the hardware configuration, including the PROFIBUS-DP gateway.
  Instructions: Software manual, chapter "Connecting to the safety controller"
- Switch to the **Gateway** view.
- Click on the byte you do not need and wish to delete.




- Click on the **Delete** icon in the command bar.

You will find further information about how to configure the process diagram here:

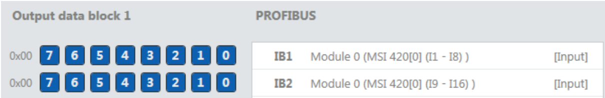
- *Configuration of gateways with MSI.designer [chapter 5]*

- Software manual

**Allocating bytes to other addresses**

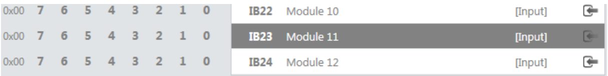
MSI.designer allocates the addresses by default. You can manually change this address allocation by moving bytes.

In our example, we have shifted **byte 1** to **byte 23** in **tab 1**.




**Step 1: Check target address**

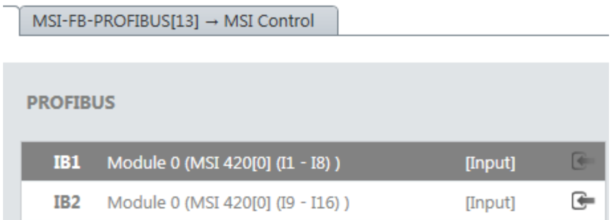
↳ Ensure that the desired address (**byte 23** in our example) has not been allocated.




↳ When the target address is assigned here, delete the bytes placed there.
To do this, click on the byte in the work area and click on the **Delete** symbol in the command bar.

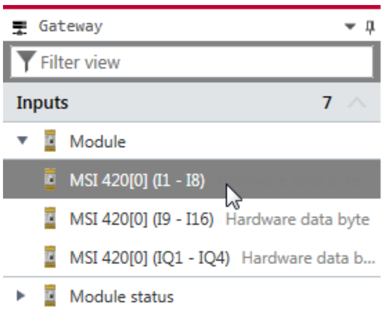
**Step 2: Delete byte from original address**

↳ Delete the byte you wish to reallocated (**byte 1** in our example).
To do this, click on the byte in the work area and click on the **Delete** symbol in the command bar.

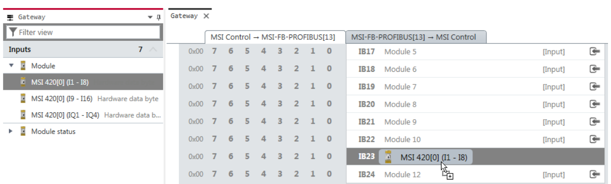



**Step 3: Place byte on new target address**

↳ Open the **Gateway** docking window and select the desired bytes under the associated module.




↳ Use the mouse button to drag the Byte into the work area on **byte 23**.



## 9.3 PROFIBUS configuration of the gateway - how the data are transferred

The following steps are required to configure communication between the PLC and the gateway.

> ***NOTICE***
>
> This documentation does not address the installation of the PROFIBUS-DP network or the other components of the automation system project in the network configuration tool. It is assumed that the PROFIBUS project in the configuration program, e.g. the SIEMENS SIMATIC Manager, has already been set up. The examples presented are based on configurations created with the help of the SIEMENS SIMATIC Manager.

### Step 1: Install the device master file (GSD)

Before the MSI-FB-PROFIBUS can be used for the first time as part of the network configuration tool, e.g. the SIEMENS SIMATIC Manager, the device master file (GSD) of the gateway must first be installed in the hardware catalog of the tool.

- ↳ Download the GSD file and the equipment symbol from the product site of the MSI-FB-PROFIBUS module (www.leuze-shop.com).
- ↳ Follow the instructions for installing GSDs in the online help section or in the user manual for the PROFINET network configuration tool.

If you are using SIEMENS SIMATIC Manager (HW Config), then the gateway then appears in the hardware catalog under >> **Additional field devices > PROFIBUS DP > Network transitions > Leuze electronic GmbH + Co. KG > Leuze > MSI >** MSI-FB-PROFIBUS

### Step 2: Add the gateway to the project

To make the system of the MSI 400 system available in the process diagram of the PLC, the gateway must first be added to the hardware configuration. The procedure to be used depends on the hardware configuration software of the PLC used. Please also read the documentation for the corresponding software in this regard.

The example below shows how the gateway is added to a SIEMENS SIMATIC manager project.

In the SIEMENS SIMATIC Manager, then you can find the gateway in the hardware catalog under >> **Additional field devices > PROFIBUS DP > Leuze electronic GmbH + Co. KG > Leuze > MSI >** MSI-FB-PROFIBUS

- ↳ Use the drag&drop function to pull the equipment into the PROFIBUS network. Example:

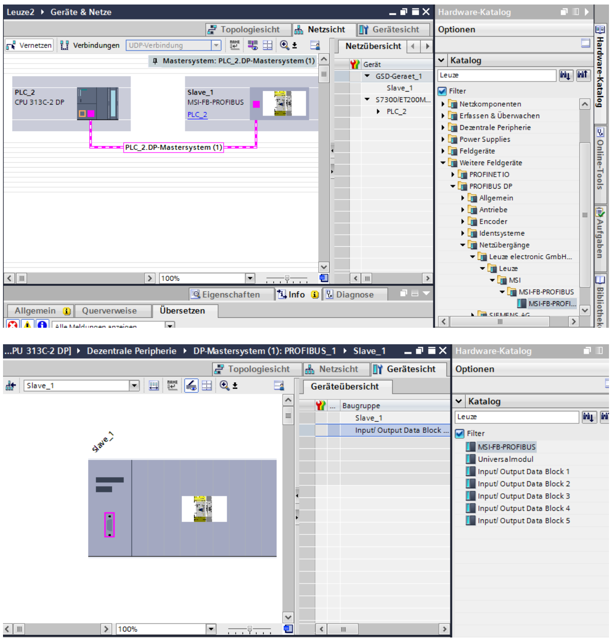



Fig. 9.5: PROFIBUS-DP gateway in the PROFIBUS HW Config

### Diagnostic data of MSI-FB-PROFIBUS module

The MSI-FB-PROFIBUS module makes diagnostic data available via PROFIBUS-Standard-DP-V0 diagnostics:

- Standard diagnosis (6 bytes)
- Device-related diagnosis State messages or manufacturer-specific messages

Each module has a unique diagnostic ID. The gateway determines the manufacturer-specific diagnosis number based on this ID. In this way, module-specific diagnosis texts can be read out of the GSD. The following table shows the content of the diagnosis messages:

Tab. 9.8: Content of the PROFIBUS diagnosis messages

| Octet | Content | Comment |
|---|---|---|
| 7 | 0x09 | Header |
| 8 | See following table | Diagnostics ID |
| 9 | 0 | PROFIBUS slot number of the module. The PROFIBUS gateway supports five slots, which do not, however, represent physical slots. For this reason, all messages should be assigned to Slot 0 (the gateway itself). |
| 10 (Bit 0…2) | 001 or 010 | 001 = Incoming error, 010 = Outgoing error |
| 10 (Bit 3…7) | 00000… 11111 | Alarm sequence number, increased on each state change of octet 10, bit 0 … 2 (incoming/outgoing error) |
| 11 | 0 ... 14 | Position of the module that caused the diagnosis message.<br>0 = controller module<br>1 = 1st I/O module<br>…<br>13 = 1st Gateway<br>14 = 2nd Gateway<br>(Relay output expansions are not counted) |
| 12 to 15 | Variable | 4 bytes with module-specific diagnosis data.<br>See below: Table “PROFIBUS error messages” |

The following picture shows the raw data output of a Profibus diagnostic message where the byte order of the module-specific diagnostic data has been specified. The data itself are the module status bits of the corresponding module (octet 11: 00 > controller module) and can be decoded with the table belonging to the module.

Tab. 9.9: Key

| Range | Description |
|---|---|
| A | See *Content of the PROFIBUS diagnosis messages [chapter 9.3]* |
| B | See *Meaning of the module status bits of the MSI 430-x/???- LEUZE controller module [chapter 3.3.4]* |

The following table shows the diagnostics IDs for the MSI 400 system:

Tab. 9.10: UnitDiagType of the MSI 400 system

| UnitDiagType (=Diagnostic ID + 160) | Modules |
|---|---|
| 161 | MSI 400 |
| 162 | safe I/O module (MSI-EM-I8, MSI-EM-IO84) |
| 163 | PROFIBUS gateway (MSI-FB-PROFIBUS) |
| 164 | CANopen gateway (MSI-FB-CANOPEN) |
| 165 | EtherCAT gateway (MSI-FB-ETHERCAT) |
| 166 | Reserved |
| 167 | Reserved |
| 168 | Reserved |
| 169 | Reserved |
| 170 | Reserved |
| 171 | Controller module 1: 32-bit state |
| 172 | Controller module 2: 32-bit state |
| 173 | Controller module 3: 32-bit state |
| 174 | unsafe I/O module (MSI-EM-IO84NP) |

The following table shows the module-specific diagnostic data (as defined in the GSD) and the corresponding error messages.

Tab. 9.11: PROFIBUS error messages

| Diagnostics ID | Diagnosis bit (X_Unit_Diag_Bit) | Error cause | Error message |
|---|---|---|---|
| 01 | 0 | Reserved | Reserved |
| | 1 | | Module operating state is Critical Error. |
| | 2 | | Power supply not in permitted range |
| | 3 | | Reserved |
| | 4 | | Configuration of a module in the system is incompatible or invalid |
| | 5 | | Power supply not in permitted range |
| | 6 | | Reserved |
| | 7 | | Communication error on EFI2 |
| | 8 to 31 | | Reserved |

| Diagnostics ID | Diagnosis bit (X_Unit_Diag_Bit) | Error cause | Error message |
|---|---|---|---|
| 11, 12, and 13 | 0 | Controller module | Reserved |
| | 1 | | Module operating state is Critical Error. |
| | 2 | | Power supply not in permitted range |
| | 3 | | Reserved |
| | 4 | | Configuration of a module in the system is in-compatible<br>or invalid |
| | 5 | | Power supply at B1 not in permitted range |
| | 6 | | Fast shut-off collective error |
| | 7 | | Power supply at B2 not in permitted range |
| | 8 to 95 | | Description of bits 8 to 959: See *Table "Mean-ing of the module state bits" [chapter 3.3.4]* |
| 2 | 0 | I/O module | Reserved |
| | 1 | | Internal error: Internal tests failed or monitor-ing test failed or poor process data or self-test failed |
| | 2 | | External error: External tests failed |
| | 3 | | Error history element exists: Access with con-figuration tool |
| | 4 | | Configuration is incompatible or invalid |
| | 5 | | Output power supply not in permitted range |
| | 6 to 7 | | Reserved |
| | 8 | | Dual-channel evaluation of input 1 - 2: Error detected |
| | 9 | | Dual-channel evaluation of input 3 - 4: Error detected |
| | 10 | | Dual-channel evaluation of input 5 - 6: Error detected |
| | 11 | | Dual-channel evaluation of input 7 - 8: Error detected |

| Diagnostics ID | Diagnosis bit (X_Unit_Diag_Bit) | Error cause | Error message |
|---|---|---|---|
| 2 | 12 | I/O module | Reserved |
| | 13 | | Reserved |
| | 14 | | Module state input data |
| | 15 | | Module state output data |
| | 16 | | Error of the external test signal at Input 1. Check for short-circuit to high or cabling error |
| | 17 | | Error of the external test signal at Input 2. Check for short-circuit to high or cabling error |
| | 18 | | Error of the external test signal at Input 3. Check for short-circuit to high or cabling error |
| | 19 | | Error of the external test signal at Input 4. Check for short-circuit to high or cabling error |
| | 20 | | Error of the external test signal at Input 5. Check for short-circuit to high or cabling error |
| | 21 | | Error of the external test signal at Input 6. Check for short-circuit to high or cabling error |
| | 22 | | Error of the external test signal at Input 7. Check for short-circuit to high or cabling error |
| 2 | 23 | I/O module | Error of the external test signal at input 8. Check for short-circuit to high or cabling error |
| | 24 | | Error: Short-circuit after high at Output 1 |
| | 25 | | Error: Short-circuit after low at Output 1 |
| | 26 | | Error: Short-circuit after high at Output 2 |
| | 27 | | Error: Short-circuit after low at Output 2 |
| | 28 | | Error: Short-circuit after high at Output 3 |
| | 29 | | Error: Short-circuit after low at Output 3 |
| | 30 | | Error: Short-circuit after high at Output 4 |
| | 31 | | Error: Short-circuit after low at Output 4 |
| 3 | 0 | PROFIBUS gateway | Reserved |
| | 1 | | Internal error: Internal tests failed |
| | 2 | | Reserved |
| | 3 | | Reserved |
| | 4 | | Configuration is incompatible or invalid |
| | 5 | | Module state input data |
| | 6 | | Module state output data |
| | 7 to 31 | | Reserved |

| Diagnostics ID | Diagnosis bit (X_Unit_Diag_Bit) | Error cause | Error message |
|---|---|---|---|
| 4 | 0 | CANopen gateway | Reserved |
| | 1 | | Internal error: Internal tests failed |
| | 2 | | Reserved |
| | 3 | | Reserved |
| | 4 | | Configuration is incompatible or invalid |
| | 5 | | Module state input data |
| | 6 | | Module state output data |
| | 7 to 31 | | Reserved |
| 5 | 0 | EtherCAT Gateway | Reserved |
| | 1 | | Internal error: Internal tests failed |
| | 2 | | Reserved |
| | 3 | | Reserved |
| | 4 | | Configuration is incompatible or invalid |
| | 5 | | Module state input data |
| | 6 | | Module state output data |
| | 7 to 31 | | Reserved |
| 6 | 0 | Other module | Reserved |
| | 1 | | Internal error: Internal tests failed |
| | 2 | | Reserved |
| | 3 | | Reserved |
| | 4 | | Configuration is incompatible or invalid |
| | 5 | | Reserved |
| | 6 | | Reserved |
| | 7 to 31 | | Reserved |
| 7 | 0 | Other module | Reserved |
| | 1 | | Internal error: Internal tests failed |
| | 2 | | Reserved |
| | 3 | | Reserved |
| | 4 | | Configuration is incompatible or invalid |
| | 5 | | Reserved |
| | 6 | | Reserved |
| | 7 to 31 | | Reserved |
| 8 | 0 | Other module | Reserved |
| | 1 | | Internal error: Internal tests failed |
| | 2 | | Reserved |
| | 3 | | Reserved |
| | 4 | | Configuration is incompatible or invalid |
| | 5 | | Reserved |
| | 6 | | Reserved |
| | 7 to 31 | | Reserved |

| Diagnostics ID | Diagnosis bit (X_Unit_Diag_Bit) | Error cause | Error message |
|---|---|---|---|
| 9 | 0 | Other module | Reserved |
| | 1 | | Internal error: Internal tests failed |
| | 2 | | Reserved |
| | 3 | | Reserved |
| | 4 | | Configuration is incompatible or invalid |
| | 5 | | Module state input data |
| | 6 | | Module state output data |
| | 7 to 31 | | Reserved |
| 10 | 0 | Other module | Reserved |
| | 1 | | Internal error: Internal tests failed |
| | 2 | | Reserved |
| | 3 | | Reserved |
| | 4 | | Configuration is incompatible or invalid |
| | 5 to 31 | | Reserved |
| 14 | 0 | Unsecure IO | Reserved |
| | 1 | | Internal error: Internal tests failed |
| | 2 | | External error: External tests failed |
| | 3 | | Reserved |
| | 4 | | Configuration is incompatible or invalid |
| | 5 | | Output power supply not in permitted range |
| | 6 to 14 | | Reserved |
| | 15 | | Module status: Output data |
| | 16 … 31 | | Reserved |

## 9.4 Diagnostics and troubleshooting

You can find information on the diagnostics of the MSI 400 system in the software manual.

Tab. 9.12: Troubleshooting on the MSI-FB-PROFIBUS module

| Error | Possible cause | Possible remedy |
|---|---|---|
| Key: ○ LED off / ✹ LED flashes / ● LED lights up | | |
| MSI.designer cannot set up a connection to the MSI 400 gateway | The MSI-FB-PROFIBUS module has no power supply | • Switch on the power supply.<br>• Check the communication settings in MSI.designer. |
| The MSI-FB-PROFIBUS module does not provide any data.<br>LED PWR: ● Green<br>LED BF: ○ Off<br>MS LED: ✹ Red (1 Hz) | • Configuration required.<br>• The configuration has not yet been fully transmitted. | • Configure the MSI-FB-PROFIBUS module and transfer the configuration to the system.<br>• Wait until the configuration has been fully transferred. |
| The MSI-FB-PROFIBUS module does not provide any data.<br>LED PWR: ● Green<br>LED BF: ○ Off<br>MS LED: ● Green | No data set was activated. | Activate at least one data set. |
| The MSI-FB-PROFIBUS module does not provide any data.<br>LED PWR: ● Green<br>LED BF: ○ / ● Off / red<br>MS LED: ✹ Green (1 Hz) | The MSI-FB-PROFIBUS module is in the Stop state. | • The controller module/application is stopped.<br>• Start the controller module (switch to Run mode). |
| The MSI-FB-PROFIBUS module does not provide any data.<br>LED PWR: ● Green<br>LED BF: ○ Off<br>MS LED: ● Green | PROFIBUS-Master is in stop mode | Set the PROFIBUS-Master to Run mode |

<table>
<tr><th colspan="2">Error</th><th>Possible cause</th><th>Possible remedy</th></tr>
<tr><td colspan="2">The MSI-FB-PROFIBUS functioned correctly after configuration but suddenly provides no more data.</td><td rowspan="4"><ul><li>The PROFIBUS hardware address of the MSI-FB-PROFIBUS module was changed.</li><li>The PROFIBUS line has been interrupted.</li></ul></td><td rowspan="4"><ul><li>Check the PROFIBUS address settings on the hardware.</li><li>Check the PROFIBUS cabling.</li><li>Check the PROFIBUS master.</li></ul></td></tr>
<tr><td>LED PWR</td><td>Green</td></tr>
<tr><td>LED BF</td><td>Red</td></tr>
<tr><td>MS LED</td><td>/<br>Red / green</td></tr>
<tr><td colspan="2">The MSI-FB-PROFIBUS module is in the Critical error state.</td><td rowspan="4"><ul><li>Internal device error on the MSI-FB-PROFIBUS module</li><li>The module version of the controller module does not support any gateways.</li></ul></td><td rowspan="4"><ul><li>Switch the MSI 400 system's power supply off and on again.</li><li>Check the diagnostic messages with the aid of the MSI.designer.</li><li>Use the controller module with the required module version.</li><li>If the error persists, replace the gateway.</li></ul></td></tr>
<tr><td>LED PWR</td><td>Green</td></tr>
<tr><td>LED BF</td><td>Red</td></tr>
<tr><td>MS LED</td><td>Red (2 Hz)</td></tr>
<tr><td colspan="2">The MSI-FB-PROFIBUS module / the MSI 400 system is in the “Critical error” state.</td><td rowspan="4"><ul><li>The MSI-FB-PROFIBUS module is not properly connected to the MSI 400 modules.</li><li>The module connection plug is dirty or damaged.</li><li>Another MSI 400 module has an internal critical error.</li></ul></td><td rowspan="4"><ul><li>Plug the MSI-FB-PROFIBUS module in correctly.</li><li>Clean the connection plug and socket.</li><li>Switch on the power supply again.</li><li>Check the other MSI 400 modules.</li></ul></td></tr>
<tr><td>LED PWR</td><td></td></tr>
<tr><td>LED BF</td><td>Off</td></tr>
<tr><td>MS LED</td><td></td></tr>
</table>

# 10 CANopen gateway

The following MSI 400 gateway can be used for CANopen:

- MSI-FB-CANOPEN

## 10.1 Interfaces and operation

### Operating and display elements

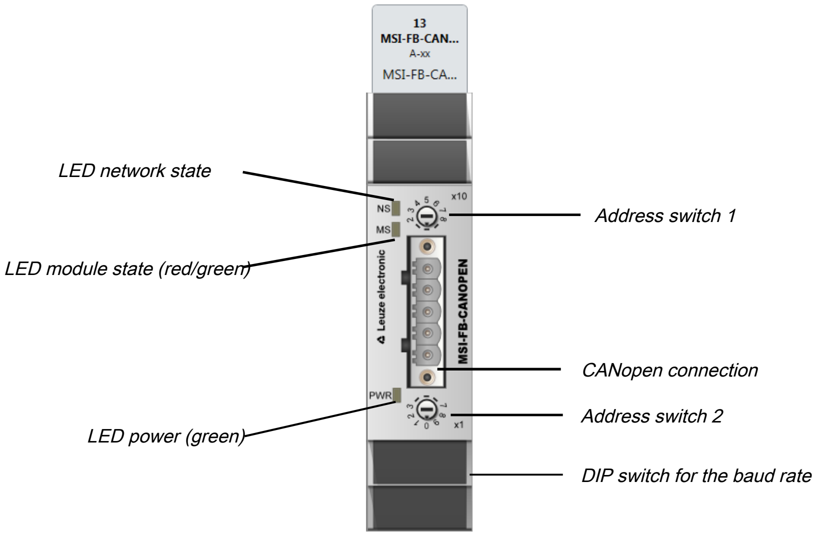



Fig. 10.1: Operating and display elements of the MSI-FB-CANOPEN module

Tab. 10.1: Reference: State LEDs of the MSI-FB-CANOPEN module

| **LED** | | **Meaning** |
|---|---|---|
| Key: ○ LED off / (flashing) LED flashes / ● LED lights up | | |
| **PWR Power** | ○ Off | No power supply |
| | ● Green | Ready for operation, power supply switched on |
| | ● Red | System error |
| **NS (Net-work status)** | ○ Off | CANopen state: stopped (except for node guarding and heartbeat, when activated) |
| | ● Green | CANopen state: Ready for operation (PDO and SDO data ex-change) |
| | (flashing) Green | CANopen state: Pre-operational (only SDO data exchange) |
| | ● Red | CAN-Bus off (hardware problem on CAN - physical layer) or error passive |
| | (flashing) Red (1 Hz) | Node guarding failed (NMT master no longer monitors the slave) or heartbeat consumer failure |

| LED | | Meaning |
|---|---|---|
| **MS (module status)** | Off | Switch on |
| | Green | Executing, internal safety bus and PDO status: all module status bits are "Good" |
| | Green (flashing) | Idle (cable not connected or node guarding failed) |
| | Red / green | Executing, internal safety bus and PDO status: At least one module status bit is “Bad”, see *Troubleshooting on the MSI-FB-CANOPEN module [chapter 10.15]*. |
| | Red | Critical error, caused by emergency bit |
| | Red (1 Hz) | Configuration required or is taking place right now |
| | Red (2 Hz) | Critical error, caused by gateway itself |

Further information: *Diagnostics and troubleshooting [chapter 10.13]*

**NOTICE**

- To allow the PLC to detect the MSI-FB-CANOPEN module as bus participant, the PLC must already be started up, before the MSI 400 system is switched on.
- If a PLC is stopped or is switched off then the MSI-FB-CANOPEN module can go into the **Error passive** or **CAN Bus Off** states. In these cases the MSI 400 system must be reset before reuse with a PLC.

### How to set the CANopen address with the aid of the hardware address switches

- Set the CANopen address switches using the hardware address switches at the front of the system.
- Switch the MSI 400 system off and on again.

Tab. 10.2: Address switch on the MSI-FB-CANOPEN module

| Switches | Function |
|---|---|
| **× 10** | Address switch 1<br>Rotary switch with 10 positions for setting the module address<br>(in tens) |
| **× 1** | Address switch 2<br>Rotary switch with 10 positions for setting the module address (in units) |

### How to set the baud rate with the aid of the hardware DIP switches:

- Set the baud rate using the DIP switches on the equipment.
- Switch the MSI 400 system off and on again.

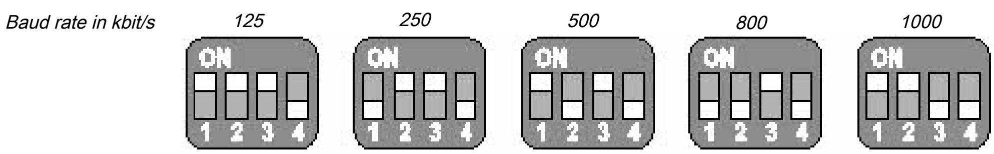



Fig. 10.2: Setting the DIP switches on the MSI-FB-CANOPEN module

Tab. 10.3: Setting the DIP switches on the MSI-FB-CANOPEN module

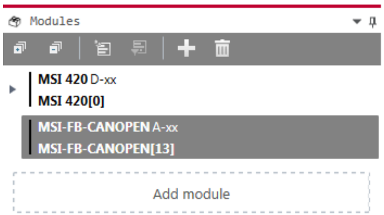
| Baud rate (kbit/s) | DIP 1 | DIP 2 | DIP 3 | DIP 4 |
|---|---|---|---|---|
| **125 (default)** | On | On | On | On |
| **125** | On | On | On | Off |
| **250** | Off | On | On | Off |
| **500** | On | Off | On | Off |
| **800** | Off | Off | On | Off |
| **1000** | On | On | Off | Off |

**NOTICE**

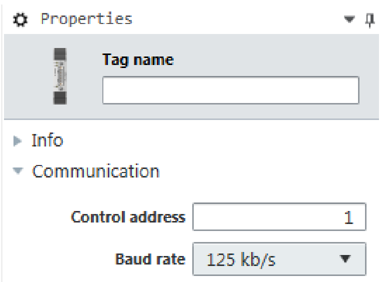
- All other DIP switch settings will set the baud rate to 125 kbit/s.
- When the address switches on the equipment are set to "00", the DIP switch settings are ignored and the baud rate setting in MSI.designer is used.

### How to set the CANopen address and the baud rate using the software

- Set the two hardware address switches on the front of the device to "00".
- Launch MSI.designer.
- Read in the hardware configuration, including the CANopen gateway.
  Instructions: Software manual, chapter "Connecting to the safety controller"
- Change to the **Modules** docking window and click the MSI-FB-CANOPEN module in the work area.




- Open the **Properties** docking window.
  - ⇨ You see the module configuration dialog.




- Under **Communication**, enter the desired values for the parameters **Controller address** and **baud rate**.
- Connect MSI.designer with the MSI 400 system and transmit the configuration.
  More information on connecting with the controller: Software manual, chapter "Connecting to the safety controller"

**NOTICE**

- You can set an address within the 1 ... 99 range with the aid of the hardware address switches.
- You can set an address within the 1 ... 127 range with the aid of the MSI.designer.
- The CAN-open master cannot overwrite the address.
- When the CANopen address and the baud rate are set with the aid of MSI.designer , the settings become valid immediately after transferring the configuration (i.e. without first switching the MSI 400 system off and on again). Exception: When the system is in the Bus-Off state, a power cycle is required.

### Pin assignment

The connection to the CANopen field bus takes place with the aid of a 5-pin open-style plug.

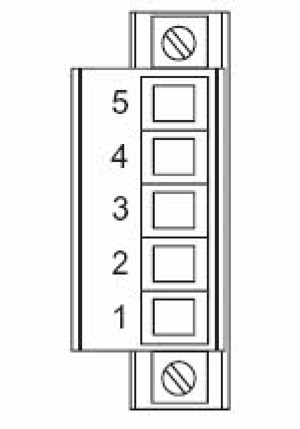



Fig. 10.3: Open-style plug on the MSI-FB-CANOPEN module

Tab. 10.4: Reference: Allocation of open-style plug on the MSI-FB-CANOPEN module

| Pin | Description | |
|---|---|---|
| 5 | – | – |
| 4 | H CAN_H | CAN High |
| 3 | DR (CAN_SHLD) | Screening connection (optional) |
| 2 | L CAN_L | CAN Low |
| 1 | – | – |

### Bus cable

CANopen is based on a linear topology with screened, two-lead twisted-pair cables and terminal resistances at both bus ends. The screening is connected to ground at both ends. The transmission rate depends on the network length and ranges from 125 kbit/s to 1000 kbit/s. The potential network lengths range from 20 m at 1000 kbit/s to 500 m at 125 kbit/s.

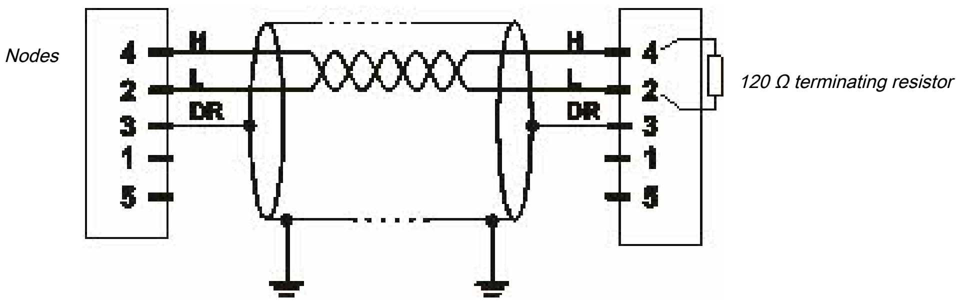



Fig. 10.4: CANopen bus cable

| | NOTICE |
|---|---|
| ! | It is not necessary to connect a power supply (Pin 1/5) to the MSI-FB-CANOPEN module. |

The following maximum physical values are possible:

Tab. 10.5: Maximum line lengths on the MSI-FB-CANOPEN module

| Bit rate (kbit/s) | Maximum cable length (m) |
|---|---|
| 125 | 500 |
| 250 | 250 |
| 500 | 100 |
| 800 | 40 |
| 1000 | 20 |

**EDS file**

The equipment characteristics are described with the aid of the electronic data sheet (EDS file), that makes use of any standard bus configuration tool.

You will fin the EDS file and the equipment symbol for integration into a PLC of the product website of the MSI-FB-CANOPEN module on the Internet (www.leuze-shop.com).

## 10.2 CANopen configuration of the gateway - how the data are transferred

| | *NOTICE* |
|---|---|
| ! | This documentation does not address the installation of the CANopen network or the other components of the automation system project in the network configuration tool. It is assumed that the CANopen project in the configuration program, e.g. 3S Software CoDeSys 2.x, has already been set up. The examples presented are based on configurations created with the help of CoDeSys 2.3. |

The following steps are required to configure communication between the PLC and the gateway.

**Step 1: Install the electronic data sheet (EDS file)**

Before the MSI-FB-CANOPEN module can be used for the first time as equipment in the network configuration tool, e.g. CoDeSys 2.3, the electronic data sheet (EDS file) of the gateway must first be installed in the hardware catalog of the tool.

- ↳ Download the EDS file and the equipment symbol from the product site of the MSI-FB-CANOPEN module (www.leuze-shop.com).
- ↳ Follow the instructions for the installation of EDS files in the online help section or in the user manual for the CANopen network configuration tool.

**Example – How to install the EDS file with CoDeSys 2.3:**

- ↳ Open the window for editing the **control configuration**.

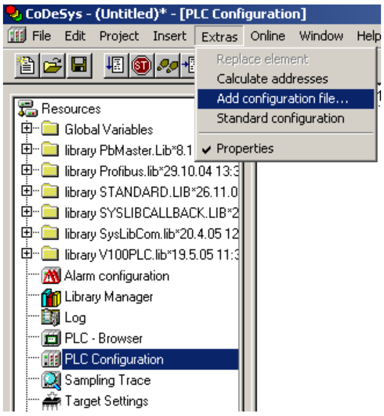



Fig. 10.5CoDeSys editing window for control configuration

- ↳ Choose the command **Add configuration file...** from the Extras menu. A file selection window is opened.
- ↳ Select the EDS file of the MSI-FB-CANOPEN module and click the **Open** button.

**Step 2: Add the gateway to the controls**

To make the system of the MSI 400 system available in the process diagram of the PLC, the gateway must first be added to the hardware configuration. The procedure to be used depends on the hardware configuration software of the PLC used. Please also read the documentation for the corresponding software in this regard.

- ↳ Open the window for editing the **control configuration** and select the controls.
- ↳ Click the controller with the right mouse button or open the **Insert** menu.

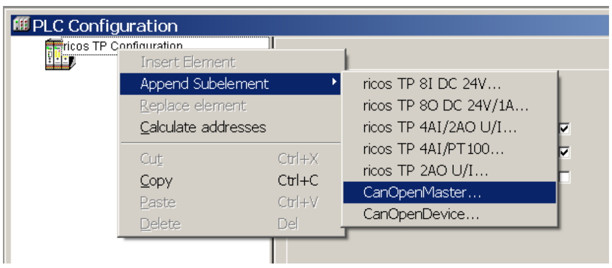



Fig. 10.6: Attaching a CanMaster with CoDeSys 2.3

- ↳ Select the command **CanMaster ....** from one of the two menus under **Attach sub-element.** A CanMaster will be attached to the controls.

↳ Now select the CanMaster.

↳ Click the CanMaster with the right mouse button or open the **Insert** menu.

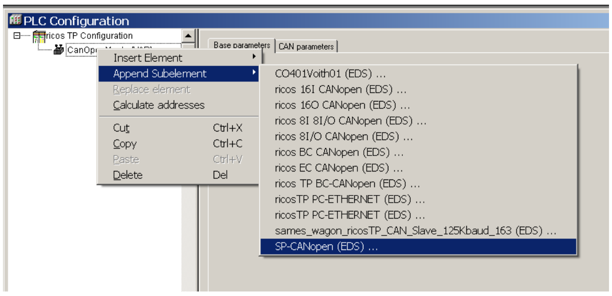


Fig. 10.7: Attaching the MSI-FB-CANOPEN module with CoDeSys 2.3

↳ In one of the two menus, under **Attach subelement**, select the command **"**MSI-FB-CANOPEN (EDS)", to attach the MSI-FB-CANOPEN module to the CanMaster.

**Step 3: Select and configure the process data objects (PDOs)**

Once you have added the device to the automation network, you must configure the process data objects to be used and how to transfer them.

**Example – How to install the PDO transmission type with CoDeSys 2.3:**

↳ In the Control Configuration edit window, select the MSI-FB-CANOPEN module. Then click the Send PDO mapping index card on the right.

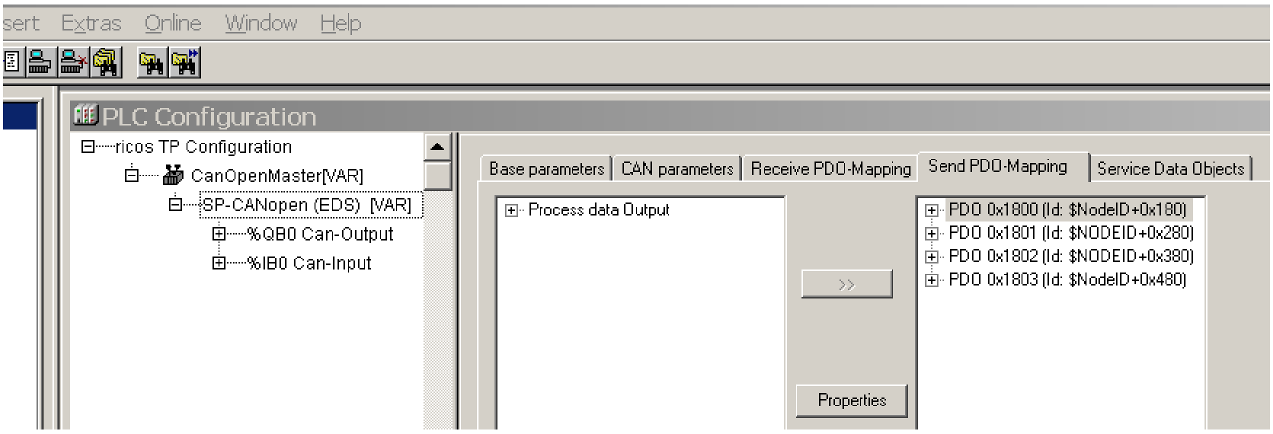



Fig. 10.8: PDO configuration with CoDeSys 2.3

↳ Select one of the PDOs shown (e.g. PDO 1) and click on the Properties button. The PDO Properties dialog window will open.

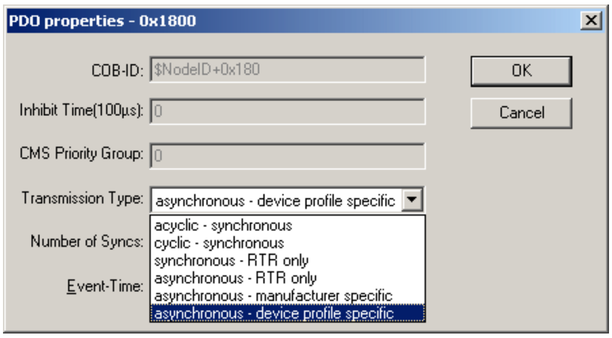



Fig. 10.9: PDO Properties dialog window in CoDeSys 2.3

↳ From the selection, choose the desired transmission type for the PDO, enter the event time in ms and click on OK. More detailed information in this regard may be found in the section "Transmission types for the TxPDOs" on page 107 and in the manual for your CanOpen configuration software.

↳ Repeat these steps for the other transmission and receiving PDOs.

## 10.3 CANopen configuration of the gateway - which data are transferred

Each CANopen device stores its data in objects listed in the object directory. The service data objects (SDOs) mainly contain the CANopen configuration data, while the process data are stored in process data objects (PDOs). Communication objects are used to read and write these SDOs and PDOs and to control the devices. The following sections contain more detailed descriptions of the various objects.

**Predefined Connection Set (PCS)**

The Predefined connection set provides a simple CAN identifier structure. The MSI-FB-CANOPEN gateway makes communication objects available, which can be contacted or transmitted using this CAN Identifier. The PCS consists of 2 broadcast objects (NMT and SYNC) and a total of 12 peer-to-peer objects. Each of these objects has a clear 11-bit CAN identifier, which consists of a function code and a device address. The device address for the broadcast objects is 0, while that for the other objects is within the range of 1 ... 127.

Tab. 10.6: Structure of the CAN identifiers

| **Bit number** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Function code | | | | Device address | | | | | | |

Tab. 10.7: PCS communication objects

| Object | CAN identifier | Meaning |
|---|---|---|
| **Broadcast objects** | | |
| **Peer-to-peer objects** | | |
| NMT | 00h | Network management |
| SYNC | 80h | Sync message |
| EMERGENCY | 081h…0FFh | State message |
| TxPDO1 | 181h…1FFh | Send process data object 1 |
| RxPDO1 | 201h…27Fh | Receive process data object 1 |
| TxPDO2 | 281h…2FFh | Send process data object 2 |
| RxPDO2 | 301h…37Fh | Receive process data object 2 |
| TxPDO3 | 381h…3FFh | Send process data object 3 |
| RxPDO3 | 401h…47Fh | Receive process data object 3 |
| TxPDO4 | 481h…4FFh | Send process data object 4 |
| RxPDO4 | 501h…57Fh | Receive process data object 4 |
| TxSDO | 581h…5FFh | Send service data project |
| RxSDO | 601h…67Fh | Receive service data object |
| NMT-ErrorControl | 701h…77Fh | Node guarding |

Each object starts with a CAN identifier, followed by a RTR bit (remote transmission request), followed by a data length code (DLC), followed by 0 to 8 data bytes. The DLC (4 bits) provides the number of data bytes.

## 10.4 NMT – network management

The broadcast object NMT is used to start, stop or initialize CANopen devices. A device in the CANopen network must take on the role of the NMT master for this purpose. This is usually the PLC. All other devices are regarded as NMT slaves. NMT services are broadcast services to which the slaves do not generate responses.

All NMT objects start with the CAN-ID 00h.

**Broadcast service for an NMT slave with the address N:**

Tab. 10.8: Network management for an NMT slave with the address N

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 00h | 2 | OP | N | | | | | | |

**Broadcast service for all NMT slaves:**

Tab. 10.9: Network management for all NMT slaves:

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 00h | 2 | OP | 0 | | | | | | |

| OP | NMT command | Explanation |
|---|---|---|
| 80h | Go to "Pre-Operational" | After booting, an NMT slave will automatically go into the pre-operational state. In this state,<br>communication via SDOs is permitted, but not via PDOs. The NMT slave can be set to this state from another state. |
| 01h | Go to "Operational" | The operational state is reached from the "pre-operational" state. Communication via PDOs is possible in this state and the CANopen slave responds to sync commands.<br>**Note:** During the transition to the NMT operational state, each slave sends a TxPDO with the transmission type = 255, so that the NMT master is informed about the current input configuration. |
| 02h | Go to "Prepared/Stopped" | Communication via SDO or PDO is not possible in this state and the device also does not respond to sync commands. |
| 81h | Go to "Reset node" | This will trigger a re-initialization of the CANopen function in the NMT slave. |
| 82h | Go to "Reset communication" | This will trigger a re-initialization of the CANopen function in the NMT slave; the toggle bit for node guarding is set to 0. |

**Example for resetting all communication:**

The following NMT object (CAN-ID = 00h) contains 2 data bytes (DLC = 2). Data byte 1 contains the command "Reset communication" (82h), data byte 2 addresses this command to all devices in the CANopen network (address = 0):

Tab. 10.10: Example of an NMT object for resetting all communication

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 00h | 2 | 82h | 0 | | | | | | |

## 10.5 SYNC

The SYNC command results in all TxPDOs of a CANopen slave being sent. It is thus possible to prompt the slave with the aid of SYNC.

Tab. 10.11: Prompting of inputs with the aid of SYNC

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 80h | 0 | | | | | | | | |

The slave sends all input values when he receives this command. All TxPDOs are sent.

To ensure that the slave automatically sends the current input values when receiving a SYNC command, the transmission type for the relevant PDOs must be set to 1 (cyclic, synchronous). In addition, the device must be in the "operational" state.

It is possible to amend the transmission type for the TxPDOs with the aid of the SDOs 1800 … 1803 (PDO communication parameter) and to amend Sub-Object 2. The following types are permitted:

- Acyclic/synchronous = 0
- Cyclic/synchronous = 1 = 1 ... 240
- Acyclic once device profile = 255 (only for TxPDO 1 … 4, digital inputs)

## 10.6 Emergency

A CANopen slave with the address N sends an emergency message to inform the other devices of an error state.

Tab. 10.12: Emergency messages

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 80h + N | 8 | ErrL | ErrH | Err-Reg | M1 | M2 | M3 | M4 | M5 |
| | | | | | | | | | |
| ErrL, ErrH | | Emergency error code, 16-bit (high-order byte (ErrH) / low-order byte (ErrL))<br>$00xx_h$: Error reset or no error<br>$10xx_h$: General error<br>…<br>$8110_h$: CAN overflow<br>$8120_h$: Passive error<br>$8130_h$: Life Guard error<br>…<br>$82xx_h$: Protocol error<br>…<br>$FFxx_h$: device-specific error, with xx as transition, see *Emergency states and transitions [chapter 10.12]*. | | | | | | | |
| Err-Reg | | Error register, CANopen object SDO 1001h | | | | | | | |
| M1 | | The higher-order half-byte contains the diagnostics ID from build state A-08.<br>The low-order half-byte contains the module index and thus corresponds to the module address of the module list and names the module causing the error.<br>The diagnostic ID can additionally be determined from the content of SDO 1027 with the subindex (= M1 + 1), whereby only the least significant half byte (=module index) may be taken for the calculation of the subindex of M1.<br>The diagnostics ID is required as an index for the "CANopen Emergency Messages" table (see below) to assign the status bits to the corresponding module. | | | | | | | |
| M2 … M5 | | 4 bytes, module-specific state bits. Active bits are high (= "1").<br>(see below: Table “CANopen Emergency Messages”) | | | | | | | |

The following table shows the module-specific diagnostic data and the corresponding error messages.

It should be noted that the diagnostic bit indicates the position of the affected bit and not the bit value itself; the bit value indicates the error case and here has the value "0", see also *Diagnostic example from CANopen Gateway module version A-08 [chapter 10.14]*.

Tab. 10.13: CANopen emergency messages

| Diag-nostics ID | Diagnos-tic bit (M5 … M 2) | Emergency cause | Emergency message |
|---|---|---|---|
| **10 [1)], 11[2)], 12[3)], 13[4)]** | **00** | Controller module | Reserved |
| | **01** | | Internal error: Internal tests failed |
| | **02** | | External error: External tests failed |
| | **03** | | Power supply at A1 not in permitted range |
| | **04** | | Configuration of a module in the system is incompatible or invalid |
| | **05** | | Power supply at B1 not in permitted range |
| | **06** | | Fast Shut-Off collective fault |
| | **07** | | Power supply at B2 not in permitted range |
| | **08 to 95** | | Description of bits 8 to 95: See *Table "Meaning of the module state bits" [chap-ter 3.3.4]* |
| | [1)] Diagnostics ID 10 relates to bit 00-31 or bit 32-63 or bit 64-95, the assignment is not unique<br>[2)] Diagnostics ID. 11 relates to bit 00-31<br>[3)] Diagnostics ID. 12 relates to bit 32-63<br>[4)] Diagnostics ID 13 relates to bit 64-95 | | |
| **02** | **00** | Secure I/O modules | Reserved |
| | **01** | | Internal error: Internal tests failed |
| | **02** | | External error: External tests failed |
| | **03** | | Error history element exists: Access with configuration tool |
| | **04** | | Configuration is incompatible or invalid |
| | **05** | | Output power supply not in permitted range |
| | **06** | | Reserved |
| | **07** | | Reserved |
| **02** | **08** | Secure I/O modules | Dual-channel evaluation of input 1-2: Er-ror detected |
| | **09** | | Dual-channel evaluation of input 3-4: Er-ror detected |

| Diag- nostics ID | Diagnos- tic bit (M5 … M 2) | Emergency cause | Emergency message |
|---|---|---|---|
| 02 | 10 | Secure I/O modules | Dual-channel evaluation of input 5-6: Error detected |
| | 11 | | Dual-channel evaluation of input 7-8: Error detected |
| | 12 | | Reserved |
| | 13 | | Reserved |
| | 14 | | Module state input data |
| | 15 | | Module state output data |
| 02 | 16 | Secure I/O modules | Error of the external test signal at Input 1. Check whether there is a short-circuit to High or a cabling error |
| | 17 | | Error of the external test signal at Input 2. Check whether there is a short-circuit to High or a cabling error |
| | 18 | | Error of the external test signal at Input 3. Check whether there is a short-circuit to High or a cabling error |
| | 19 | | Error of the external test signal at Input 4. Check whether there is a short-circuit to High or a cabling error |
| | 20 | | Error of the external test signal at Input 5. Check whether there is a short-circuit to High or a cabling error |
| | 21 | | Error of the external test signal at Input 6. Check whether there is a short-circuit to High or a cabling error |
| 02 | 22 | Secure I/O modules | Error of the external test signal at input 7. Check whether there is a short-circuit to High or a cabling error |
| | 23 | | Error of the external test signal at Input 8. Check whether there is a short-circuit to High or a cabling error |
| 02 | 24 | Secure I/O modules | Error: Short-circuit after high at Output 1 |
| | 25 | | Error: Short-circuit after low at Output 1 |
| | 26 | | Error: Short-circuit after high at Output 2 |
| | 27 | | Error: Short-circuit after low at Output 2 |
| | 28 | | Error: Short-circuit after high at Output 3 |
| | 29 | | Error: Short-circuit after low at Output 3 |
| | 30 | | Error: Short-circuit after high at Output 4 |
| | 31 | | Error: Short-circuit after low at Output 4 |

| Diag-nostics ID | Diagnos-tic bit (M5 … M 2) | Emergency cause | Emergency message |
|---|---|---|---|
| 03 | 00 | PROFIBUS gateway | Reserved |
| | 01 | | Internal error: Internal tests failed |
| | 02 | | Reserved |
| | 03 | | Reserved |
| | 04 | | Configuration is incompatible or invalid |
| | 05 | | Module state input data |
| | 06 | | Module state output data |
| | 07 to 31 | | Reserved |
| 04 | 00 | CANopen gateway | Reserved |
| | 01 | | Internal error: Internal tests failed |
| | 02 | | Reserved |
| | 03 | | Reserved |
| | 04 | | Configuration is incompatible or invalid |
| | 05 | | Module state input data |
| | 06 | | Module state output data |
| | 07 to 31 | | Reserved |
| 05 | 00 | EtherCAT gateway | Reserved |
| | 01 | | Internal error: Internal tests failed |
| | 02 | | Reserved |
| | 03 | | Reserved |
| | 04 | | Configuration is incompatible or invalid |
| | 05 | | Module state input data |
| | 06 | | Module state output data |
| | 07 to 31 | | Reserved |
| 06 | 00 | Reserved | Reserved |
| | 01 | | Internal error: Internal tests failed |
| | 02 | | Reserved |
| | 03 | | Reserved |
| | 04 | | Configuration is incompatible or invalid |
| | 05 to 31 | | Reserved |
| 07 | 00 | Reserved | Reserved |
| | 01 | | Internal error: Internal tests failed |
| | 02 | | Reserved |
| | 03 | | Reserved |
| | 04 | | Configuration is incompatible or invalid |
| | 05 to 31 | | Reserved |

| Diag-nostics ID | Diagnos-tic bit (M5 … M 2) | Emergency cause | Emergency message |
|---|---|---|---|
| 08 | 00 | Reserved | Reserved |
| | 01 | | Internal error: Internal tests failed |
| | 02 | | Reserved |
| | 03 | | Reserved |
| | 04 | | Configuration is incompatible or invalid |
| | 05 to 31 | | Reserved |
| 09 | 00 | Reserved (other module) | Reserved |
| | 01 | | Internal error: Internal tests failed |
| | 02 | | Reserved |
| | 03 | | Reserved |
| | 04 | | Configuration is incompatible or invalid |
| | 05 | | Module state input data |
| | 06 | | Module state output data |
| | 07 to 31 | | Reserved |
| 14 | 00 | Unsecure IO | Reserved |
| | 01 | | Internal error: Internal tests failed |
| | 02 | | External error: External tests failed |
| | 03 | | Reserved |
| | 04 | | Configuration is incompatible or invalid |
| | 05 | | Output power supply not in permitted range |
| | 06 … 14 | | Reserved |
| | 15 | | Module status: Output data |
| | 16 … 31 | | Reserved |

***NOTICE***

The allocation of the diagnostic bits for M2 to M5 is as follows:

| Bit 0 | Bit 1 | … | Bit 7 | Bit 8 | … | Bit31 |
|---|---|---|---|---|---|---|
| M5.0 | M5.1 | … | M5.7 | M4.0 | … | M2.7 |

**See also**

Diagnostic example from CANopen Gateway module version A-08 [ 155]

## 10.7 Node guarding

An NMT master (e.g. a PLC with integrated CANopen master) uses the NMT-Error-Control object to detect a failure of an NMT slave with the

address N. The NMT slave must respond to the query of the NMT master within the node guarding time. The node guarding time must be monitored by the NMT master.

The NMT master sends a CAN message with the identifier <700h + node ID> and RTRBit (remote transmission request).

Query of NMT master:

Tab. 10.14: Query of NMT master

| CAN-ID | RTR | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 700h + N | 1 | 0 | | | | | | | | |

The slave (e.g. the MSI-FB-CANOPEN module) then sends a state byte 1 with the following content:

Response of the slave:

Tab. 10.15: Response of the slave

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 700h + N | 1 | Byte1 | | | | | | | |

Tab. 10.16: Remote Transmission Request

| Bit | Meaning | |
|---|---|---|
| 7 | Toggle bit changes its value between two consecutive queries | |
| 6…0 | NMT status | 4 = Stopped<br>5 = Operational<br>127 = Pre-operational |

### Bootup

On booting, the gateway sends a bootup message with the CAN-ID 700h+N, DLC = 1 and byte 1 = 0.

### Heartbeat producer

When the gateway has been configured as a heartbeat producer (i.e. when SDO 1017 contains a value for the producer heartbeat time, see table *"Supported SDOs" [chapter 10.10]*), then sends a cyclical message with the CAN-ID 700h+N, DLC = 1 and Byte 1 = 05h. The toggle bit (bit 7) is always 0.

### Heartbeat consumer

When the gateway has been configured as a heartbeat consumer (i.e. when SDO 1016.1 contains a value for the consumer heartbeat time, see table *"Supported SDOs" [chapter 10.10]*), then at least one node guarding message must be received within the configured consumer heartbeat time (typically from a NMT master).

## 10.8 PDO communication

Process data objects (PDOs) are the real-time objects of the CANopen field bus. They are sent without a protocol overhead, i.e. the receiver sends no confirmation.

The MSI-FB-CANOPEN module provides four Transmit process data objects (TxPDOs) that contain the process data to be sent to the network and four Receive process data objects (RxPDOs) for the process data to be received from the network.

CANopen objects are addressed with the aid of 11-bit CAN identifiers. As a pre-set, the CAN identifier derives each object from the object type and the configured CANopen device address. The CAN identifier of the PDOs can be changed by using SDOs 1400 to 1403 for the RxPDOs and SDOs 1800 to 1803 for the TxPDOs ("PDO linking").

| | NOTICE |
|---|---|
| ! | Each process data object contains 8 bytes. |

The content of the process data objects can be freely selected, but has been preconfigured as follows in MSI.designer:

Tab. 10.17: Preset for the content of the transmit process data objects (TxPDOs) of the MSI-FB-CANOPEN module

| | **PDO#1** | **PDO#2** | **PDO#3** | **PDO#4** |
|---|---|---|---|---|
| | **Output data - Block 1** | **Output data - Block 2** | **Output data - Block 3** | **Output data - Block 4** |
| **Byte 0** | Input values for Module 0 (I1..I8) | Input values for Module 1 | Input values for Module 9 | Output values for Module 5 |
| **Byte 1** | Input values for Module 0 (I9..I16) | Input values for Module 2 | Input values for Module 10 | Output values for Module 6 |
| **Byte 2** | Input values for Module 0 (IQ1..IQ4) | Input values for Module 3 | Input values for Module 11 | Output values for Module 7 |
| **Byte 3** | Output values for Module 0 (Q1..Q4,IQ1-IQ4) | Input values for Module 4 | Input values for Module 12 | Output values for Module 8 |
| **Byte 4** | Direct data (Off) 1 | Input values for Module 5 | Output values for Module 1 | Output values for Module 9 |
| **Byte 5** | Direct data (Off) 2 | Input values for Module 6 | Output values for Module 2 | Output values for Module 10 |
| **Byte 6** | Direct data (Off) 3 | Input values for Module 7 | Output values for Module 3 | Output values for Module 11 |
| **Byte 7** | Direct data (Off) 4 | Input values for Module 8 | Output values for Module 4 | Output values for Module 12 |

Detailed information about the content of the process diagram may be found here: *Configuring the gateway output values (tab 1) [chapter 5.3]*

You will find further information about how to configure the process diagram here:

- *Configuration of gateways with MSI.designer [chapter 5]*
- Software manual

**NOTICE**

- The process data can also be written and read with the aid of service data objects SDO 6000 and SDO 6200 (see *SDO communication [chapter 10.9]*). Easy access via SDO is recommended for diagnostic purposes. More rapid PDO communication is to be used for normal operation.
- After starting up or changing the configuration (either with the aid of the CANopen master or with MSI.designer), the LED MS of the CANopen gateway flashes red/green until an initial transmit/receive data exchange has taken place via PDO or SDO 6000/SDO 6200 in the CANopen network.

**TxPDO 1…4**

A transmit-PDO transmits data from the CANopen gateway to a CANopen device.

Tab. 10.18: TxPDO 1…4

| CAN ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 181-1FF | 8 | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 |
| 281-2FF | 8 | B9 | B10 | B11 | B12 | B13 | B14 | B15 | B16 |
| 381-3FF | 8 | B17 | B18 | B19 | B20 | B21 | B22 | B23 | B24 |
| 481-4FF | 8 | B25 | B26 | B27 | B28 | B29 | B30 | B31 | B32 |

**B1…B32:** CAN telegram bytes as in the network input data, with the aid of MSI.designer (see *Configuring the gateway output values (tab 1) [chapter 5.3]*).

The gateway sends one or several TxPDOs when at least one of the following events occurs:

- At least one input or output byte has changed its value and the transmission type for the TxPDO that contains this byte has the value 255.
- At least one input or output byte has changed its value and the gateway contains a SYNC command and at least one TxPDO has transmission type 0.
- When the transmission type is n = 1 ... 240, n sync commands are required in order to send the TxPDO.
- The transmission type for a TxPDO is 254 or 255 and the event timer (SDO 1800,5 for TxPDO1) has a value of N > 0. In this case this TxPDO is sent every N ms.
- A TxPDO can also be called up with the aid of a remote transmission request (RTR). This requires a CAN telegram to the gateway that contains the CAN-ID of the desired TxPDOs with DLC = 0 and RTR = 1.

The operating state of the device must be "operational" for all transmission methods (see *Table "Network management for all NMT slaves" [chapter 10.4]*).

**RxPDO 1…4**

A receive-PDO transmits data from a CANopen device to the CANopen gateway.

Tab. 10.19: RxPDO 1…4

| CAN ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 201-1FF | 8 | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 |
| 301-2FF | 8 | B9 | B10 | B11 | B12 | B13 | B14 | B15 | B16 |
| 401-3FF | 8 | B17 | B18 | B19 | B20 | B21 | B22 | B23 | B24 |
| 501-4FF | 8 | B25 | B26 | B27 | B28 | B29 | B30 | B31 | B32 |

**B1…B32:** CAN telegram bytes as for the gateway input data, with the aid of MSI.designer.

The transmission type 255 is preset for all RxPDOs. This means that the gateway immediately transmits the RxPDO data on to the controller module. This setting cannot be changed.

## 10.9 SDO communication

SDOs are service data objects. They contain a wide spectrum of different data. This includes configuration as well as input and output data.

Contrary to PDO communication, the receipt of each SDO is answered at protocol level,

i.e. the receiving device sends a confirmation.

This CANopen PCS implementation supports the following protocols:

- SDO Download Expedited (write SDO)
- SDO Upload Expedited (read SDO)
- Upload SDO Segment Protocol (segmented reading of an SDO)

**SDO Download Expedited (write SDO)**

The client sends a request to server N. The 16-bit index and the sub-index for the SDO to be written form part of this message. In addition, the request contains 4 data bytes with the data to be written.

Tab. 10.20: Write SDO

| CAN ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 600h + N | 8 | 23h | SDO_L | SDO_H | SUB | Byte 1 | Byte 2 | Byte 3 | Byte 4 |

SDO_L = SDO-Index, Low Byte

SDO_H = SDO-Index, High Byte

SUB = SDO-Subindex

The server then responds with a confirmation:

Tab. 10.21: SDO write confirmation

| CAN ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 580h + N | 8 | 60h | SDO_L | SDO_H | SUB | Byte 1 | Byte 2 | Byte 3 | Byte 4 |

Byte 1 to 4 in the write confirmation contain zeros.

**SDO Upload Expedited (read SDO)**

The client requests the content of an SDO by submitting a request to server N. The 16-bit index and the sub-index for the SDO to be read form part of this message. Byte 1 to 4 in the read request contain zeros.

Tab. 10.22: Read SDO

| CAN ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 600h + N | 8 | 40h | SDO_L | SDO_H | SUB | Byte 1 | Byte 2 | Byte 3 | Byte 4 |

The server responds with the following message. Bytes 1 to 4 contain the value of the requested object.

Tab. 10.23: SDO read confirmation

| CAN ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 580h + N | 8 | 42h | SDO_L | SDO_H | SUB | Byte 1 | Byte 2 | Byte 3 | Byte 4 |

### The CANopen data types UDINT and UINT

In order to transmit the data types UDINT or UINT, the data must be in Intel format. For example, the 32-bit value 12345678h in data bytes 5, 6, 7 and 8 must be transmitted in the following order: [5] = 78, [6] = 56, [7] = 34, [8] = 12.

**NOTICE**

This also applies to the SDO index in data bytes 2 and 3, which is of the data type UINT. This means that the low byte is transmitted in data byte 2 and the high byte in data type 3.

**Example:** The following messages are required to read SDO 1003,1 of the CANopen device with device address 2. The data type of the data to be read is UDINT.

The client sends:

| CAN ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 602h | 8 | 40h | 03h | 10h | 01h | 00h | 00h | 00h | 00h |

The server responds:

| CAN ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 582h | 8 | 42h | 03h | 10h | 01h | 08h | 00h | 50h | 02h |

The combined response data result in the 32-bit word 02500008h.

## 10.10 SDO object directory

Each CANopen device manages its SDOs in an object directory. The complete object directory is formally described in an EDS file. Many CANopen tools can ready this EDS file and therefore know the object characteristics of the CANopen device.

The following table shows all SDOs for the MSI-FB-CANOPEN gateway.

Tab. 10.24: Supported SDOs

| SDO # | Type |
|---|---|
| 1000 | Device type |
| 1001 | Error register |
| 1003 | Error list (error history) |
| 1005 | COB ID SYNC |
| 1008 | Device name |
| 1009 | Hardware version |
| 100A | Software version |
| 100C | Guard Time |
| 100D | Life Time Factor |
| 1016 | Consumer Heartbeat Time |
| 1017 | Producer Heartbeat Time |
| 1018 | Identification |
| 1027 | Module list |
| 1400…1403 | Communication parameter for RxPDO 1 … 4 |
| 1600…1603 | Mapping parameter for RxPDO 1 … 4 |
| 1800…1803 | Communication parameter for TxPDO 1 … 4 |
| 1A00…1A03 | Mapping parameter for TxPDO 1 … 4 |
| 3100 | Module state bits |
| 3200 | Project CRC |
| 3300 | Module type code |
| 6000 | Process data input objects |
| 6200 | Process data output objects |

You will find more detailed information about these SDOs in the CANopen standard draft DS 301 V4.02 (DSP 301 V4.1).

**SDO 1001: Error register**

The error register is a bit field of 8 bits and indicates the type of error if one of the subsequent bit positions is set to "1".

Tab. 10.25: Unsupported error register values

| Bit position | Meaning |
|---|---|
| 0 | "generic error" |
| 4 | "communication error" |
| 7 | "communication error" |

**SDO 1003: Error list (error history)**

SDO 1003 is an array that contains the last 10 error codes that the gateway has reported with the aid of emergency messages. Array index 0 contains the number of error codes recorded in SDO 1003.

A new error is recorded in index 1, while older errors will in this case be renumbered (incremented by 1). The array index can be overwritten with a 0 from the outside, thus completely deleting the array.

| | NOTICE |
|---|---|
| | ↳ Not all errors reported with the aid of emergency messages are recorded in SDO 1003, only the errors listed here: *Error and state information for the modules [chapter 3.3.4]*and table "*CANopen Emergency Messages [chapter 10.6]*"<br>↳ The entries in SDO 1003 are in UDINT format and normally divided into 16 bits of error code and 16 bits of additional information. In the event of an emergency, the module state diagnosis (4 bytes) will be entered here. |

**SDO 1005: COB ID SYNC**

SDO 1005 contains the COB-ID of the sync object. This value has been preset to 80h, but can be changed.

| | NOTICE |
|---|---|
| | When you change the COB-ID of the sync object, please ensure that the new ID has not already been allocated to another communication object. |

**SDO 1008: Device name**

SD0 1008 contains a device name (VISIBLE STRING).

| | NOTICE |
|---|---|
| | This SDO cannot be read with a simple "SDO upload expedited". The "Upload SDO segment protocol" command (client command code ccs = 3) must be used instead, as described in the CANopen specifications DS 301. |

**SDO 1009: Hardware version**

SDO 1009 contains the current hardware version of the device (VISIBLE STRING).

| | NOTICE |
|---|---|
| | This SDO cannot be read with a simple "SDO upload expedited". The "Upload SDO segment protocol" command (client command code ccs = 3) must be used instead, as described in the CANopen specifications DS 301. |

**SDO 100A: Software version**

SDO 100A contains the current software version of the device (VISIBLE STRING).

| | NOTICE |
|---|---|
| | This SDO cannot be read with a simple "SDO upload expedited". The "Upload SDO segment protocol" command (client command code ccs = 3) must be used instead, as described in the CANopen specifications DS 301. |

**SDO 100C: Guard Time**

The guard time (UINT) multiplied by the life time factor (SINT) results in the life guarding time.

**Life Guarding Time [ms] = Guard Time [ms] × Life Time Factor**

During the Life Guarding Time, the master must send at least one node guarding message to the slave. When the life guarding time is exceeded (life guarding error), the gateway reports a cable break error and sets all network process data to 0; the LED NS starts to flash red.

In the slave, life guarding is activated by the first node guarding message when the life guarding time has not been set to 0. When the guard time or the life time factor are set to 0 after activating life guarding, life guarding will be deactivated.

Also see: *Guarding protocols [chapter 10.11]*.

### SDO 100D: Life Time Factor

SDO 100D contains the Life Time Factor (SINT). See SDO 100C.

| | NOTICE |
|---|---|
| | The Life Time Factor must either be = 0 (deactivated) or V 1.5. |

### SDO 1016: Consumer Heartbeat Time

The gateway is configured as a heartbeat consumer if SDO 1016 contains a value greater than 0 for the consumer heartbeat time. The consumer heartbeat time is given in ms.

The NMT master must send at least one node guarding message to the slave within this time. When the consumer heartbeat time is exceeded (life guarding error), the gateway reports a cable break error and sets all network process data to 0; the LED NS starts to flash red.

### SDO 1017: Producer Heartbeat Time

The gateway can also act as a heartbeat producer, i.e. send a heartbeat signal.

This allows another device to detect whether the heartbeat producer (i.e. the gateway) is still functioning correctly.

The producer heartbeat time is given in ms. For internal processing, it is rounded up to the next higher multiple of 4. If the heartbeat time is set to 0, the heartbeat signal is deactivated.

The heartbeat signal consists of a cyclic CAN message with the identifier 700h + device address.

| | NOTICE |
|---|---|
| | It is not possible to use heartbeat signals and life guarding messages simultaneously, as both functions make use of the same CAN identifier. |

Also see: *Guarding protocols [chapter 10.11]*

### SDO 1018: Identification

This SDO contains basic information about the gateway.

Tab. 10.26: Content of SDO 1018

| Subindex | Mapping | Format | Description |
|---|---|---|---|
| 1 | Manufacturer ID | UDINT | Unique manufacturer identification number (e.g. Leuze electronic) |
| 2 | Product description | UDINT | Device variant |
| 3 | Revision number | UDINT | Software version of the device |
| 4 | Serial number | UDINT | Serial number of the device |

Example for reading out the revision number and the serial number:

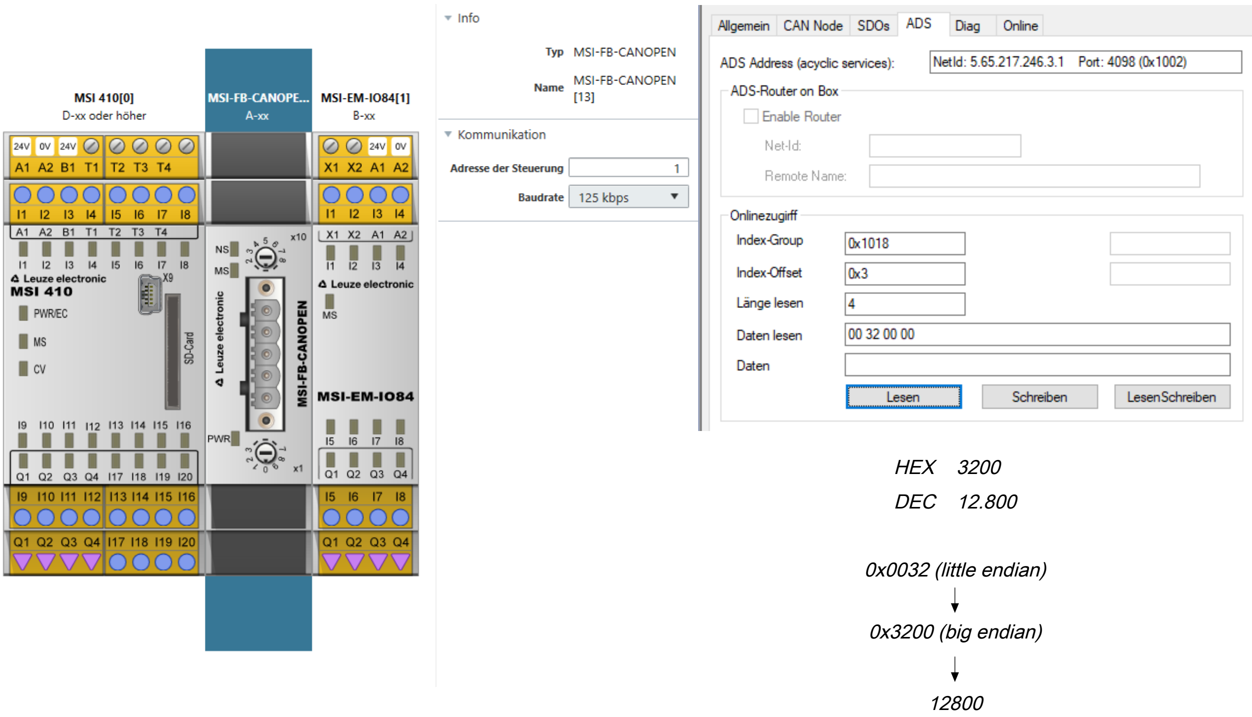



Fig. 10.10: SDO 1018: Subindex 3

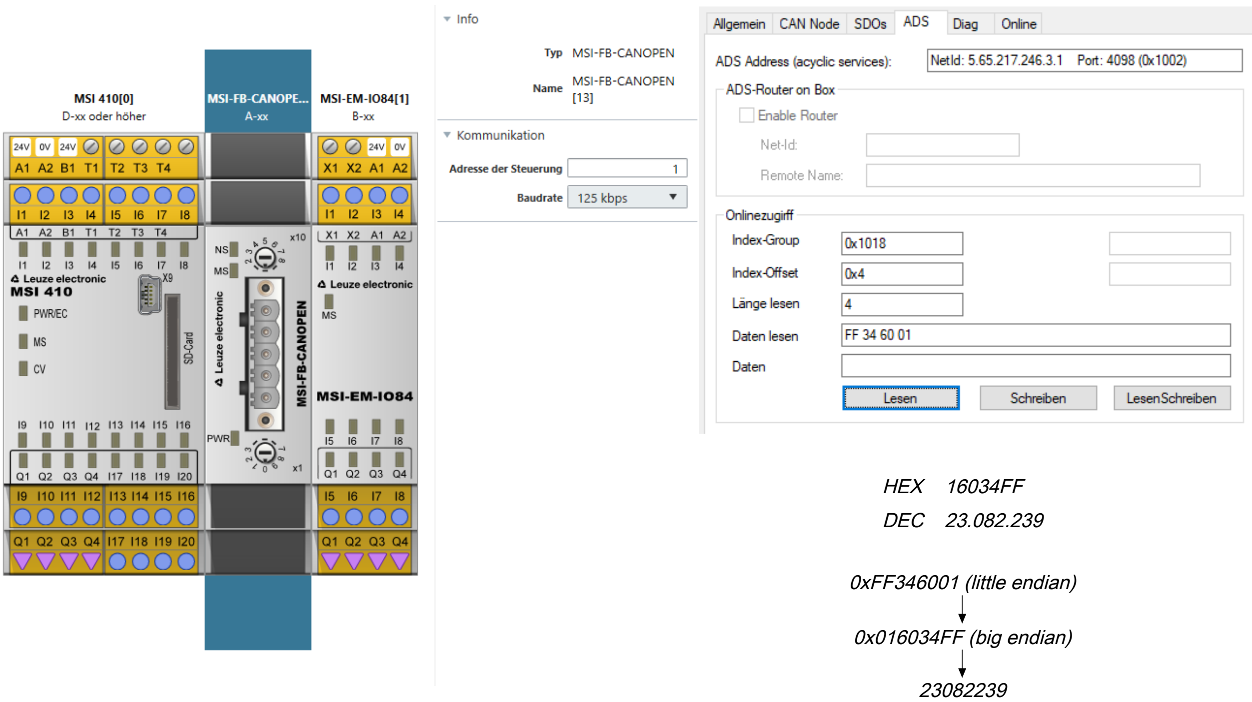



Fig. 10.11: SDO 1018: Subindex 4

**SDO 1027: Module list**

The module list contains the module type and diagnostics ID (module ID) of all safe MSI 400 modules in the system.

Example:

| |
|---|
| Subindex = 03 -> 0x00000602, whereby:<br>02 = diagnostics ID[1]<br>06 = module type[2] |
| Further information:<br>[1] See: Table *"CANopen Emergency Messages" [chapter 10.6]*<br>[2] See below: Table "Module types" |

Tab. 10.27: Content of SDO 1027

| Subindex | Meaning | Format |
|---|---|---|
| 0 | SDO 1027 entries | SINT |
| 1…15 | Module slot positions | SINT |

Tab. 10.28: Module types

| Subindex | Module type |
|---|---|
| 0 | MSI 410 (CPU without Ethernet) |
| 1 | MSI 420-x (CPU with Ethernet) |
| 2 | MSI 430-x/???- LEUZE (CPU with Modbus/TCP, PROFINET IO, EtherNet/IP) |
| 4 | MSI-EM-I8 (secure input module) |
| 6 | MSI-EM-IO84 (secure I/O module) |
| 7 | PROFIBUS DP gateway |
| 9 | CANopen gateway |
| 14 | MSI-EM-IO84NP (non-secure I/O module) |
| 22 | EtherCAT gateway |

**SDO 1400 … 1403: Communication parameters for the RxPDOs**

SDO 1400 to 1403 can be used to configure the communication parameters for RxPDOs 1 to 4, e.g. SDO 1400 defines the parameters for RxPDO 1, etc.

Tab. 10.29: Content of SDO 1400 ... 1403

| Subindex | Mapping | Format | Description |
|---|---|---|---|
| 1 | COB ID | UDINT | CAN identifier for this PDO,<br>write-protected |
| 2 | Receive mode | SINT | Fix 255 (asynchronous mode) |

The receive mode (read/write) determines how the PDO is to be received. For RxPDOs, the receive mode has been set to 255 (asynchronous mode). In this mode, the data of a RxPDOs received are directly routed to the outputs.

***NOTICE***

When the receive mode is set to a value other than 255, an error code is generated (abort code 0609 0030h, invalid parameter value).

### SDO 1600 … 1603: Mapping parameters for the RxPDOs

This SDO cannot be used, as mapping of the RxPDOs takes place with the aid of the MSI.designer. Also see: Table *"Pre-set for the content of the transmit process data objects (TxPDOs)" [chapter 10.8]*

### SDO 1800 … 1803: Communication parameters for the TxPDOs

SDO 1800 to 1803 can be used to configure the communication parameters for TxPDOs 1 to 4, e.g. SDO 1800 defines the parameters for TxPDO 1, etc.

Tab. 10.30: Content of SDO 1800 ... 1803

| Subindex | Mapping | Format | Description |
|---|---|---|---|
| 1 | COB ID | UDINT | CAN identifier for this PDO,<br>write-protected |
| 2 | Transmission type | SINT | Determines when the PDO should be sent |
| 5 | Event timer | UINT | in ms |

The transmission type for all TxPDOs to 255 (asynchronous mode, event-driven) has been preset.

The event timer contains the time in Ms for the cyclic transmission of the TxPDOs.

### Transmission types for the TxPDOs

Tab. 10.31: Transmission types for the TxPDOs

| TxPDO | Synchronous | Asynchronous | RTR |
|---|---|---|---|
| 1, 2, 3, 4 | 0, 1…240 | 254, 255 | 253 |

**NOTICE**

When the transmission type is set to an invalid value, an error code is generated (abort code 0030 0030h, invalid parameter value).

**Synchronous:** Synchronous transmission mode 0 means that the TxPDO is sent after receiving a Sync command, but only if data has changed. The synchronous transmission types n = 1 ... 240 mean that the TxPDO is sent after the nth Sync command is received.

**Asynchronous, event-driven by timer:** The asynchronous transmission type 254 (with a configured event timer) means that the TxPDO is sent each time when the event timer has expired. For example, a value of 500 for the event timer means that the gateway sends the respective TxPDO every 500 ms.

**Asynchronous, event-driven with change of state:** The asynchronous transmission mode 255 (without configured event timer) means that the TxPDO is sent each time at least one input bit contained in this PDO has changed.

**Asynchronous, event-controlled by timer or status change:** The asynchronous transmission type 255 (with a configured event timer) means that the TxPDO is sent each time when the event timer has expired or at least one input bit has changed. For example, a value of 500 for the event timer means that the gateway sends the respective TxPDO at least every 500 ms or in case of a change.

**RTR, on request:** Transmission type 253 means that the TxPDO can be requested with the aid of an RTR (remote transmission request). This requires a CAN message to the gateway with DLC = 0, RTR = 1 and the COB-ID of the TxPDO. The gateway then responds with the requested TxPDO.

### SDO 1A00 … 1A03: Mapping parameters for the TxPDOs

This SDO cannot be used, as mapping of the TxPDOs takes place with the aid of the MSI.designer.

Also see: Table "*Pre-set for the content of the transmit process data objects (TxPDOs) [chapter 10.8]*"

### SDO 3100: Module state bits

SDO 3100 contains the module state bits of the MSI 400 system (see *Table "CANopen Emergency Messages" [chapter 10.6]*). Active bits are low (= "0").

Tab. 10.32: Content of SDO 3100

| SDO array | Data set parameters | Module | Size |
|---|---|---|---|
| 3100.1-3 | Status of Module 0 | Controller mod-ule | UDINT |
| 3100.4 | Status of Module 1 | Expansion | UDINT |
| … | … | … | … |
| 3100.14 | Status of Module 11 | Expansion | UDINT |
| 3100.15 | Status of Module 12 | Expansion | UDINT |

**NOTICE**

The positions of the modules are numbered in the MSI.designer from 0 to 14. Thus the sub-index for SDO 3100 = Position + 3, with the first three sub-indices for the MSI 4xx module being used.

SDO 3100 can only be read.

### SDO 3200: Project CRC, internal CRC, reserved

Tab. 10.33: Content of SDO 3200

| SDO array | Data set parameters | Size |
|---|---|---|
| 3200.1 | Project CRC | UDINT |
| 3200.2 | System CRC[1)] | UDINT |
| 3200.3 | Reserved (EFI ACR CRC) | UDINT |

[1)] The use of the internal CRC in data set 2 is only permitted for diagnostic purposes so that Leuze Technical Support can provide further assistance.

### SDO 6000: Process data input objects

The 32 bytes of the process input data can be written into SDO array 6000. These are the same data as in RxPDO 1-4 (see *PDO Communication [chapter 10.8]*). The mapping is as follows:

Tab. 10.34: Mapping table for SDO 6000 – RxPDO 1-4

| SDO 6000 | RxPDO |
|---|---|
| 6000.1 | RxPDO 1, Byte 1 |
| … | … |
| 6000.8 | RxPDO 1, Byte 8 |
| 6000.9-16 | RxPDO 2, Byte 1-8 |
| 6000.17-24 | RxPDO 3, Byte 1-8 |
| 6000.25-32 | RxPDO 4, Byte 1-8 |

SDO 6000 can only be written.

### SDO 6200: Process data output objects

The 32 bytes of the process output data can be written into SDO array 6200. These are the same data as in TxPDO 1-4 (see *PDO Communication [chapter 10.8]*). The mapping is as follows:

Tab. 10.35: Mapping table for SDO 6200 – TxPDO 1-4

| SDO 6200 | TxPDO |
|---|---|
| 6200.1 | TxPDO 1, Byte 1 |
| … | … |
| 6200.8 | TxPDO 1, Byte 8 |
| 6200.9-16 | TxPDO 2, Byte 1-8 |
| 6200.17-24 | TxPDO 3, Byte 1-8 |
| 6200.25-32 | TxPDO 4, Byte 1-8 |

SDO 6200 can only be read.

## 10.11 Guarding protocols

CANopen offers several possibilities for active monitoring of the correct function of the field bus interface (e.g. cable break detection).

| | WARNING |
|---|---|
| | **Always use either node guarding or heartbeat!**<br>Guarding is compulsory according to the CIA CANopen specifications DS 301. Please always active either node guarding or heartbeat. When no guarding has been configured, the MSI 400 system cannot detect an interruption of the CANopen communication, for example an interrupted network cable. In this case the input and output data of the CANopen gateway may "freeze". |

### Heartbeat

A heartbeat producer is a CANopen device that sends a cyclic heartbeat message. This makes it possible for all other CANopen devices to detect whether the heartbeat producer still functions correctly and what its current status is. Heartbeat messages are transmitted at regular intervals, the Producer Heartbeat Time, which may be configured with the aid of SDO 1017. The configured 16-bit value is rounded up to the next higher multiple of 4 ms.

A heartbeat consumer is a CANopen device that expects a cyclic node guarding message within a certain time interval, i.e. the consumer heartbeat time, which can be configured with the aid of SDO 1016. If the heartbeat consumer does not receive a node guarding message within the configured consumer heartbeat time, it sends a life guarding emergency message and sets the process input data to 0. In addition, the gateway sends a "cable break" error message that can be processed by the controller module.

### Node guarding

Node guarding is carried out by a NMT master. This can be any CANopen device that can fulfill this function as a client. The NMT master sends a cyclic node guarding message to the device to be monitored, which must respond within a certain time, which is monitored by the NMT master. If the device to be monitored does not respond within the node guarding time, the NMT master treats this as a malfunction of the device and takes the corresponding actions.

### Life Guarding

Life guarding is carried out by the gateway itself. In the gateway, the life guarding time is calculated from the values of SDO 100C (guard time) and SDO 100D (life time factor). If the gateway does not receive a node guarding message from an NMT master once within this life guarding time, the gateway sends an internal "cable break" error message, which can be processed by the controller module, and the LED NS starts to flash red.

**NOTICE**

- The gateway can detect a cable break when life guarding has been activated, i.e. when both SDO 100C and SDO 100D have a value not equal to 0. In this case, Life Guarding starts as soon as the first Node Guarding request is received from an NMT master and ends when the master sends the “Reset Communication” command.
- Alternatively cable break detection is possible when the gateway has been configured as a heartbeat consumer. In this case, the cable break detection is carried out by the gateway itself.
- Heartbeat (producer) works without node guarding. In this case gateway cannot detect a cable break on the field bus.
- Heartbeat and node guarding / life guarding cannot be simultaneously used.
- If the configuration has been changed in such a way that life guarding is deactivated or activated, the entire MSI 400 system must be restarted, so that the CANopen network communication can again be correctly established.

The following table provides an overview of the supported guarding protocols, depending on the configuration of SDO 1016 and SDO 1017 (heartbeat), SDO 100C (guard time) and SDO 100D (life time factor).

Tab. 10.36: Overview and comparison of the guarding protocols

| SDO 1016 | SDO 1017 | SDO 100C × 100D | Heartbeat gateway | Life Guarding Gateway | Node guarding NMT master |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Not permitted: Always make use of either node guarding or heartbeat! | | |
| 0 | 0 | > 0 | Deactivated | **Cable break detection** | **Required** |
| > 0 | 0 | 0 | **Cyclic heartbeat (consumer)** | **Cable break detection** | **Possible for other slaves** |
| 0 | > 0 | 0 | **Cyclic heartbeat (producer)** | Not possible | Not possible, but guarding as a heartbeat consumer is possible |
| > 0 | > 0 | 0 | **Cyclic heartbeat (producer und consumer)** | Cable break detection | Not possible |
| > 0 | > 0 | > 0 | **Not permitted** | | |

**NOTICE**

It does not make sense to use heartbeat and life guarding simultaneously.

## 10.12 Error objects

The MSI-FB-CANOPEN module reports CAN-specific errors (e.g. initialization errors, cable brackets, CAN communication errors) to the controller module as internal safety bus errors.

### Emergency object

The emergency producer (CANopen gateway) sends the emergency object to the emergency consumer (any CANopen device, usually the controller) when CAN-specific errors occur or an error state occurs, as described in the table *"CANopen Emergency Messages" [chapter 10.6]*.

The emergency object is sent as described in DS 301 (CANopen specifications) in accordance with the following table:

Tab. 10.37: Emergency states and transitions

| Emergency state Before | Transition | Module-specific alarms | Emergency state After |
|---|---|---|---|
| Error-free | 1 | Incoming error | Error occurred |
| Error occurred | 2 | Error removed, other errors pending | Error occurred |
| Error occurred | 3 | Incoming error, other errors pending | Error occurred |
| Error occurred | 4 | All errors removed | Error-free |

The gateway is in one of two possible emergency states, either *error-free* or *errors detected*. Emergency objects are sent, depending on the transitions between these two emergency states. The error code in the emergency object shows the emergency state in which the gateway currently is (also see table below).

### Overview of error objects

Tab. 10.38: CAN-specific errors

| Error | Internal safety bus error code | Error type | Emergency error code<br>Error register<br>M1…M5 | Error history SDO 1003 | Results/possible remedy |
|---|---|---|---|---|---|
| CAN data overflow<br>CAN control overflow in Rx Fifo | 0x4501 | Warning | 0x8110<br>0x11<br>1, 0, 0, 0, 0 | – | • CAN messages have been lost.<br>• Limited band width.<br>• Check the CAN settings, increase the baud rate, reduce the number of participants or the data volume. |
| CAN-error-passive CAN control takes place in an error-passive state | 0x4503 | Warning | 0x8120<br>0x11<br>0, 0, 0, 0, 0 | – | The gateway is only sending recessive bits, i.e. it is invalidating its own messages.<br>The cause is either a hardware fault on the gateway or an external malfunction of the data transmission.<br>• Check the cabling. |
| CAN bus off<br>The CAN controls are in the bus off state | 0x4504 | Warning | – | – | Major transmission error. The CAN controls have separated the connection to the bus. Possible hardware defect.<br>• Switch the MSI 400 system off and on again. |

| Error | Internal safety bus error code | Error type | Emergency error code<br>Error register<br>M1…M5 | Error history SDO 1003 | Results/possible remedy |
|---|---|---|---|---|---|
| CAN-Tx-Fifo overflow<br>The CAN controls have no transmission resources | 0x4506 | Warning | 0x8110<br>0x11<br>2, 0, 0, 0, 0 | – | CAN messages that were to be sent from the gateway have been lost. The number of events for which the gateway is to send CAN messages is too high for the set baud rate.<br>• Increase the baud rate or change the configuration of the gateway. |
| CAN initialization failed.<br>The CAN controls could not be initialized | 0xC507 | Critical | – | – | The CAN controls or the transceiver may be defective.<br>• Replace the MSI-FB-CANOPEN module with a new device. |
| CANopen Life Guarding<br>CANopen Life Guarding has found a cable break | 0x4508 | Warning | 0x8130<br>0x11<br>0, 0, 0, 0, 0 | – | The gateway has generated a life guarding error message: Either an error has occurred on the node guarding or the heartbeat NMT master or the CAN cable has been interrupted.<br>• Check the CANopen master.<br>• Check the cabling. |

Tab. 10.39: Module-specific alarms (device-specific error - OxFFxx)

| Alarm | Internal safety bus error code | Emergency state transition | Emergency error code<br>Error register<br>M1…M5 | Error history SDO 1003 | Further information |
|---|---|---|---|---|---|
| Gateway detects incoming error according to trigger conditions | – | 1 | 0xFF01<br>0x81<br>M1 = Module index<br>M2…M5 = Module diagnostic data | M2, M3, M4, M5 | See *Table "CANopen Emergency Messages" [chapter 10.6]* |
| Gateway detects outgoing error, other errors exist | – | 2 | 0xFF02<br>0x81<br>M1 = Module index<br>M2…M5 = Module diagnostic data | M2, M3, M4, M5 | |
| Gateway detects incoming error, other errors exist | – | 3 | 0xFF03<br>0x81<br>M1 = Module index<br>M2…M5 = Module diagnostic data | M2, M3, M4, M5 | |
| All errors removed | – | 4 | 0x0000<br>0x00<br>M1 = 0<br>M2…M5 = 0 | – | |

## 10.13 CANopen diagnostic examples

**Example 1: Secure IO module in position 3, output Q4 has a short-circuit to high**

The gateway sends am Emergency message (see Table "*CANopen Emergency Messages [chapter 10.6]*").

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 08C | 8 | 03 | FF | 01 | 03 | 40 | 00 | 00 | 00 |

The CANopen address of the gateway is 12 (= C Hex). The secure IO module has position 1 in the MSI 400 system.

08C: Identifier (80 + C)
8: Data length code: This is followed by 8 bytes
03FF: Error code FF03: Device-specific error
01: Error register 01 of SDO 1001H
03: Module index M1: Module in position 3
40: Module state bit 30 (bit 6 of byte M2) = 1: Short-circuit to high at output 4 (see *Table "CANopen Emergency Messages" [chapter 10.6]*)

**Reading the current module status bits from SDO 3100:**

PLC requests:

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 60C | 8 | 40 | 00 | 31 | 04 | 00 | 00 | 00 | 00 |

60C: Identifier (600 + C)
8: Data length code: This is followed by 8 bytes
40: Expedited upload requirement
00 31: Index 3100
04: Subindex: Module in Position 1 (module position = subindex – 3)
(See table *"Content of SDO 3100" [chapter 10.10]*)

Gateway response:

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 58C | 8 | 42 | 00 | 31 | 04 | BF | FF | FF | FB |

58C: Identifier (580 + C)
8: Data length code: This is followed by 8 bytes
42: Upload response, size of data set is not shown
00 31: Index 3100
04: Subindex: Module in Position 1 (module position = subindex – 3)
(See table *"Content of SDO 3100" [chapter 10.10]*)
FB: Error byte M5, Bit 2 = 0: external error
BF: Error byte M2, Bit 30 = 0. Error: Short-circuit after high at Output 4

**Reading of error from the error history in SDO 1003:**

PLC requests:

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 60C | 8 | 40 | 03 | 10 | 01 | 00 | 00 | 00 | 00 |

60C: Identifier (600 + C)
8: Data length code: This is followed by 8 bytes
40: Expedited upload requirement
03 10: Index 1003
01: Sub-index: last error

Gateway response:

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 58C | 8 | 42 | 03 | 10 | 01 | 40 | 00 | 00 | 00 |

58C: Identifier (580 + C)
8: Data length code: This is followed by 8 bytes
42: Upload response, size of data set is not shown
03 10: Index 1003
01: Sub-index: last error
40: Module state bit 30 (bit 6 of byte M2) = 0: Short-circuit after high at Output 4

**Example 2: Secure I/O module with error at two-channel input I1/I2**

The gateway sends am Emergency message (see *Table “Emergency Messages [chapter 10.6]*").

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 08C | 8 | 03 | FF | 01 | 0B | 00 | 00 | 01 | 00 |

The CANopen address of the gateway is 12 (= C Hex). The MSI-EM-I8 module has position 11 in the MSI 400 system.

08C: Identifier (80 + C)
8: Data length code: This is followed by 8 bytes.
03FF: Error code FF03: Device-specific error
01: Error register 01 of SDO 1001H
0B: Module index M1: Module in position 11 (B Hex)
01: Module status bit 8 (bit 0 of byte M4) = 1: dual channel evaluation of inputs 1–2: Error detected (see *Table "CANopen Emergency Messages" [chapter 10.6]*)

**Reading the current module status bits from SDO 3100:**

PLC requests:

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 60C | 8 | 40 | 00 | 31 | 0F | 00 | 00 | 00 | 00 |

60C: Identifier (600 + C)
8: Data length code: This is followed by 8 bytes
40: Expedited upload requirement
00 31: Index 3100
0F: Subindex 0F = Module to position 12 (module position = subindex – 3)
(see also table *"Content of SDO 3100" [chapter 10.10]*)

Gateway response:

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 58C | 8 | 42 | 00 | 31 | 0F | FF | FF | FE | FB |

58C: Identifier (580 + C)
8: Data length code: This is followed by 8 bytes
42: Upload response, size of data set is not shown
00 31: Index 3100
04: Subindex: Module in Position 1 (module position = subindex – 3)
(See table *"Content of SDO 3100" [chapter 10.10]*)
FB: Error byte M5, Bit 2 = 0: external error
FE: Error byte M4, bit 0 = 0: two-channel evaluation of inputs 1–2: Error detected
(See *Table "CANopen Emergency Messages" [chapter 10.6]*)

### Reading of error from the error history in SDO 1003:

PLC requests:

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 60C | 8 | 40 | 03 | 10 | 01 | 00 | 00 | 00 | 00 |

60C: Identifier (600 + C)
8: Data length code: This is followed by 8 bytes
40: Expedited upload requirement
03 10: Index 1003
01: Sub-index: last error

Gateway response:

| CAN-ID | DLC | DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 58C | 8 | 42 | 03 | 10 | 01 | 00 | 00 | 01 | 00 |

58C: Identifier (580 + C)
8: Data length code: This is followed by 8 bytes
42: Upload response, size of data set is not shown
03 10: Index 1003
01: Sub-index: last error
01: Module status bit 8 (bit 0 of byte M4) = 0: two-channel evaluation of inputs 1–2: Error detected

## 10.14 Diagnostic example from CANopen Gateway module version A-08

### Example of emergency message: Dual channel evaluation of inputs I1/I2 not OK

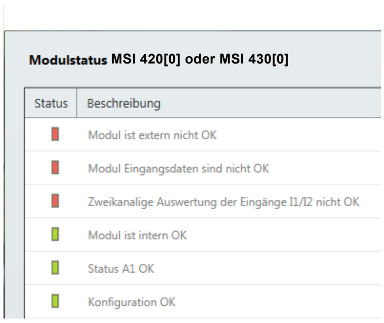



Fig. 10.12: Module status of the error in MSI.designer

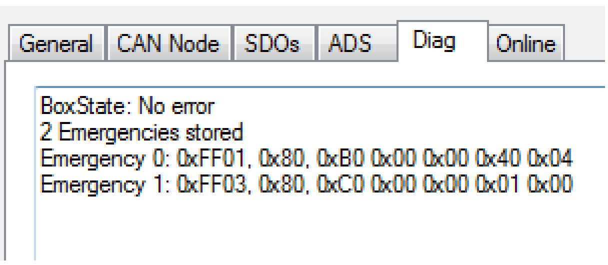



Fig. 10.13: Emergency message from the diagnostics of a PLC

Tab. 10.40: Decoding of the Emergency 0 message

| | | | |
|---|---|---|---|
| ErrL, ErrH | 0xFF01 | Gateway detects incoming error according to trigger conditions | See *Table "Module specific alarms" [chapter 10.12]* |
| Err-Reg | 0x80 | Error register corresponds to SDO 1001:00 "80" 7-bit high: Manufacturer: Specific | See *Table "Availability of data sets 1-4" [chapter 3.3]* |
| M1 | 0xB0 | Diagnostics ID 11 (B): Bit 00 – 31 (Byte 0 – 3)<br>Module index: 0 | See *Table "Emergency Messages" [chapter 10.6]* |
| M2 | 0x00 | Diagnostic bit 24 – 31 (Byte 3): – | See *Table "Emergency Messages" [chapter 10.6]*<br>See *Table "CANopen Emergency Messages" [chapter 10.6]*<br>See *Table "Meaning of module state bits of controller module (only for Modbus)" [chapter 3.3.4]* MSI 4xx |
| M3 | 0x00 | Diagnostic bit 16 – 23 (Byte 2): – | |
| M4 | 0x40 | Diagnostic bit 8 – 15 (Byte 1):<br>Module state input data | |
| M5 | 0x04 | Diagnostic bit 0 – 7 (Byte 0):<br>External module status | |

Tab. 10.41: Decoding of the Emergency 1 message

| | | | |
|---|---|---|---|
| ErrL, ErrH | 0xFF03 | Gateway detects incoming error, other errors exist | See *Table "Module specific alarms" [chapter 10.12]* |
| Err-Reg | 0x80 | Error register corresponds to SDO 1001:00 "80" 7-bit high: Manufacturer: Specific | See *Table "Availability of data sets 1-4" [chapter 3.3]* |
| M1 | 0xC0 | Diagnostics ID 12 (B): Bit 32 – 63,<br>Module index: 0 | See *Table "Emergency Messages" [chapter 10.6]* |
| M2 | 0x00 | Diagnostic bit 56 – 63 (Byte 7): – | See *Table "Emergency Messages" [chapter 10.6]*<br>See *Table "CANopen Emergency Messages" [chapter 10.6]*<br>See *Table "Meaning of module status bits of controller module (only for Modbus)"/ [chapter 3.3.4]*MSI 430-x/???-LEUZE |
| M3 | 0x00 | Diagnostic bit 48 – 55 (Byte 6): – | |
| M4 | 0x01 | Diagnostic bit 40 – 47 (Byte 5):<br>I1/I2 dual channel status | |
| M5 | 0x00 | Diagnostic bit 32 – 39 (Byte 4): – | |

## 10.15 Diagnostics and troubleshooting

You can find information about the diagnostics of the MSI 400 system in the software manual.

Tab. 10.42: Troubleshooting on the MSI-FB-CANOPEN module

<table>
<tr><th colspan="2">Error</th><th>Possible cause</th><th>Possible remedy</th></tr>
<tr><td colspan="4">Key: ○ LED off / ✴ LED flashes / ● LED lights up</td></tr>
<tr><td colspan="2">The MSI-FB-CANOPEN module does not provide any data</td><td rowspan="4">• Configuration required, node guarding or heart-beat message was not sent.<br>• The configuration has not yet been fully trans-mitted.</td><td rowspan="4">• Configure the MSI-FB-CANOPEN module and transfer the configura-tion to the system.<br>• Wait until the configura-tion has been fully trans-ferred.</td></tr>
<tr><td>LED PWR</td><td>● Green</td></tr>
<tr><td>LED NS</td><td>○ Off</td></tr>
<tr><td>MS LED</td><td>✴ Red (1 Hz)</td></tr>
<tr><td colspan="2">The MSI-FB-CANOPEN module does not provide any data</td><td rowspan="4">The configuration has not yet been fully transmitted.</td><td rowspan="4">Wait until the configuration has been fully transferred.</td></tr>
<tr><td>LED PWR</td><td>● Green</td></tr>
<tr><td>LED NS</td><td>● Green</td></tr>
<tr><td>MS LED</td><td>✴ Red (1 Hz)</td></tr>
<tr><td colspan="2">The MSI-FB-CANOPEN module does not provide any data</td><td rowspan="4">No PDO transfer since switch-on.</td><td rowspan="4">• Start the PDO transfer.[1)]<br>• Transfer the PDO via SDO 6000 or SDO 6200.</td></tr>
<tr><td>LED PWR</td><td>● Green</td></tr>
<tr><td>LED NS</td><td>● Green</td></tr>
<tr><td>MS LED</td><td>✴ / ✴ Red / green</td></tr>
</table>

| Error | Possible cause | Possible remedy |
|---|---|---|
| The MSI-FB-CANOPEN module does not provide any data<br>LED PWR: Green<br>LED NS: Green (flashing)<br>MS LED: Red / green (flashing) | • No PDO transfer since switch-on.<br>• Wrong baud rate (CAN transceiver possibly in error passive).<br>• Wrong node ID or CANopen address.<br>• The CAN cable was interrupted. | • Start the PDO transfer.[1)]<br>• Transfer the PDO via SDO 6000 or SDO 6200.<br>• Check and correct the baud rate.<br>• Check and correct the address.<br>• Check the CANopen cabling.<br>• Check the EDS file for validity and use the matching EDS file for the build status (ProductNumber and RevisionNumber parameters must match SDO content 1018sub2 and 1018sub3 respectively, or disable the corresponding test in the PLC). |
| The MSI-FB-CANOPEN module does not provide any PDO data.<br>LED PWR: Green<br>LED NS: Off /I Red / I Green<br>MS LED: Green (1 Hz) | • The MSI-FB-CANOPEN module is in the Idle state.<br>• Node guarding or heartbeat messages are sent.<br>• The MSI 400 configuration has not been verified and the controller module has been stopped. | • The controller module/ application is stopped.<br>• Start the controller module (switch to Run mode).<br>• Verify the configuration with the MSI.designer and start the controller module. |
| The MSI-FB-CANOPEN module does not provide any PDO data.<br>LED PWR: Green<br>LED NS: Green<br>MS LED: Off | Supply voltage too low. | Check the power supply. |
| The MSI-FB-CANOPEN module does not provide any data.<br>LED PWR: Red<br>LED NS: Red<br>MS LED: Red | Brief drop in power supply. | • Check the power supply.<br>• Reset the MSI 400 system. |

<table>
<tr><th colspan="2">Error</th><th>Possible cause</th><th>Possible remedy</th></tr>
<tr><td colspan="2">The MSI-FB-CANOPEN module does not provide any data.</td><td rowspan="4">• Wrong node ID or CANopen address.<br>• Wrong baud rate (CAN transceiver possibly in error passive), the MSI-FB-CANOPEN module is in idle state.</td><td rowspan="4">• Check and correct the address.<br>• Check and correct the baud rate.</td></tr>
<tr><td>LED PWR</td><td>Green</td></tr>
<tr><td>LED NS</td><td>Green (1 Hz)</td></tr>
<tr><td>MS LED</td><td>Green (1 Hz)</td></tr>
<tr><td colspan="2">The MSI-FB-CANOPEN module does not provide any data.</td><td rowspan="4">• Wrong baud rate and the transceiver of the MSI-FB-CANOPEN module is in bus-off state (hardware problem at the physical CAN level).<br>• The CAN cable was interrupted.</td><td rowspan="4">• Check and correct the baud rate.<br>• Check the CANopen cabling.<br>• Reset the MSI 400 system.</td></tr>
<tr><td>LED PWR</td><td>Green</td></tr>
<tr><td>LED NS</td><td>Red</td></tr>
<tr><td>MS LED</td><td>/<br>Red / green</td></tr>
<tr><td colspan="2">The MSI-FB-CANOPEN module does not provide any data</td><td rowspan="4">• CANopen master is in the stop or pre-operational state<br>• Another slave could not be initialized during initialization of the bus system.<br>• CANopen state of the MSI-FB-CANOPEN module is pre-operational. Wrong node ID or CANopen address.</td><td rowspan="4">• Set the CANopen master to the run state (CANopen state operational).<br>• Check whether all slaves on the bus have been switched on.<br>• Check the CANopen cabling.<br>• Check whether the CAN master starts automatically.<br>• Check and correct the CANopen address.</td></tr>
<tr><td>LED PWR</td><td>Green</td></tr>
<tr><td>LED NS</td><td>Green (1 Hz)</td></tr>
<tr><td>MS LED</td><td>Green</td></tr>
<tr><td colspan="2">The MSI-FB-CANOPEN module does not provide any data</td><td rowspan="4">• The transceiver of the MSI-FB-CANOPEN module is in the Error Passive state.<br>• The CAN cable was interrupted.</td><td rowspan="4">• Check the CANopen cabling.<br>• Check the diagnostic messages with the aid of the MSI.designer.<br>• Reset the MSI 400 system.</td></tr>
<tr><td>LED PWR</td><td>Green</td></tr>
<tr><td>LED NS</td><td>Red</td></tr>
<tr><td>MS LED</td><td>Green</td></tr>
</table>

<table>
<tr><th colspan="2">Error</th><th>Possible cause</th><th>Possible remedy</th></tr>
<tr><td colspan="2">The MSI-FB-CANOPEN module does not provide any data</td><td rowspan="4">• Node guarding or heartbeat consumer failure<br>• The guarding configuration was changed.</td><td rowspan="4">• Check the CANopen cabling.<br>• Check the life guarding time (life time factor V 1).<br>• Check the heartbeat consumer time (should be V 1.5 × heartbeat producer time).<br>• Check the diagnostic messages with the aid of the MSI.designer.<br>• Reset the MSI 400 system.</td></tr>
<tr><td>LED PWR</td><td>Green</td></tr>
<tr><td>LED NS</td><td>Red (1 Hz)</td></tr>
<tr><td>MS LED</td><td>/<br>Red / green</td></tr>
<tr><td colspan="2">The MSI-FB-CANOPEN module is in the critical error state</td><td rowspan="4">• Internal equipment error on the MSI-FB-CANOPEN module.<br>• The module version of the controller module does not support MSI 400 gateways.</td><td rowspan="4">• Switch the MSI 400 system's power supply off and on again.<br>• Check the diagnostic messages with the aid of the MSI.designer.<br>• Use the controller module with the required module version.<br>• If the error persists, replace the gateway.</td></tr>
<tr><td>LED PWR</td><td>Green</td></tr>
<tr><td>LED NS</td><td>Red</td></tr>
<tr><td>MS LED</td><td>Red (2 Hz)</td></tr>
<tr><td colspan="2">The MSI-FB-CANOPEN / the MSI 400 system is in the Critical error state.</td><td rowspan="4">• The MSI-FB-CANOPEN module is not properly connected to the MSI 400 modules.<br>• The module connection plug is dirty or damaged.<br>• Another MSI 400 module has an internal critical error.</td><td rowspan="4">• Plug the MSI-FB-CANOPEN module in correctly.<br>• Clean the connection plug and socket.<br>• Switch on the power supply<br>• once again.<br>• Check the other MSI 400 modules.</td></tr>
<tr><td>LED PWR</td><td>Red</td></tr>
<tr><td>LED NS</td><td>Off</td></tr>
<tr><td>MS LED</td><td>Red</td></tr>
<tr><td colspan="4">[1)] Configure at least one sensor/actuator so that a bit is "active" in the CAN output data.<br>Send at least one output bit to the CAN bus via the ext. PLC, so that at least one CAN input data bit is "active".<br>Info: Check that the appropriate input data bits and output data bits have been configured</td></tr>
</table>

# 11 EtherCAT gateway

The MSI 400 EtherCAT gateway can only be used in combination with controller modules of module version C-xx or higher.

### Configuration example

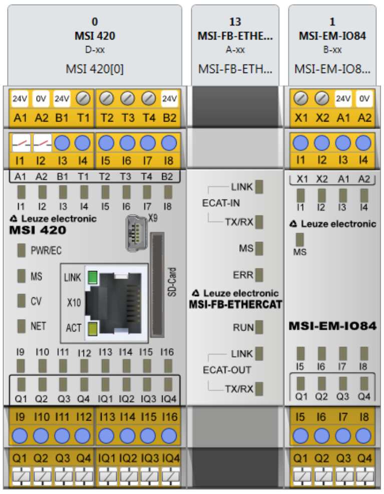



Fig. 11.1Configuration example: ???- LEUZE (0), MSI-FB-ETHERCAT (13), MSI-EM-IO84 (1)

## 11.1 Interfaces and operation

### Operating and display elements

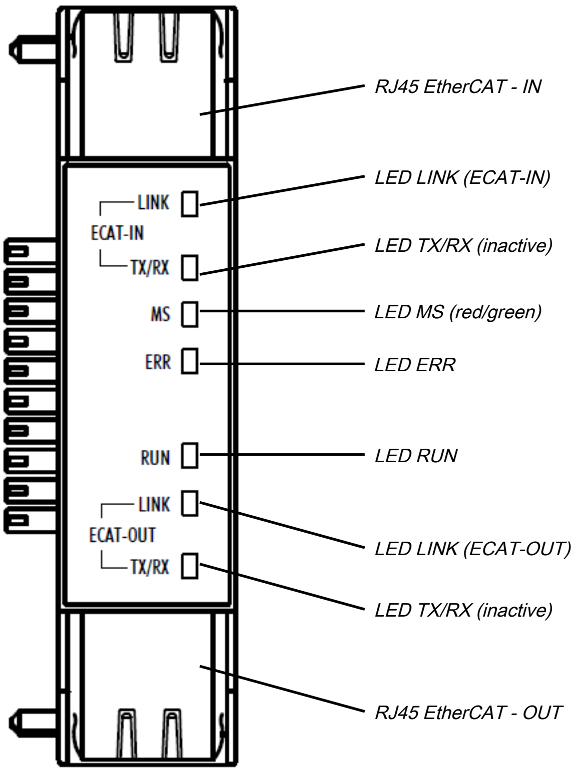



Fig. 11.2: Operating and display elements of the MSI-FB-ETHERCAT module

Tab. 11.1: Meaning of the state LEDs on the MSI-FB-ETHERCAT module

| LED | | Meaning |
|---|---|---|
| Key: ○ LED off / ✱ LED flashes / ● LED lights up | | |
| **ECAT-IN** | | |
| LINK | ○ Off | No EtherCAT device connected, no connection. |
| | ● Green | EtherCAT device connected. |
| | ✱ Green | Communication with connected EtherCAT device |
| TX/RX | ○ Off | Not used |
| **MS** | ○ Off | No voltage supply /<br>No connection to the head-end station |
| | ● Green | On:<br>MSI 400 system in operation. |
| | ✱ Green | Flashing 1 Hz:<br>MSI 400 system stopped |
| | ✱ / ✱<br>Red / green | Alternate flashing:<br>Run but the gateway has an error<br>(e.g. no EtherCAT connection) |
| | ✱ Red | Flashing 1 Hz:<br>Configuration required or is taking place right now |
| | ● Red | On:<br>Critical error |
| **ERR** | ○ Off | **No error:** The EtherCAT communication of the device is in operation |
| | ✱ Red | **Double flash**<br>**Application watchdog timeout:** An application watchdog timeout occurred<br>(Example: Sync Manager watchdog timeout) |
| | ✱ Red | **Single flash**<br>**Unrequested status change:** The slave device appli-cation has autonomously changed the EtherCAT sta-tus: The "Change" parameter in the ALStatus register is 0x01:change/error. |
| | ✱ Red | **Blink**<br>**Invalid configuration:** General configuration error<br>(Example: The configuration has not yet been fully transmitted.) |
| | ● Red | **On**<br>**Watchdog timeout:** A watchdog timeout has occurred.<br>(Example: The application controller is no longer re-sponding<br>) |

| LED | | Meaning |
|---|---|---|
| **RUN** | Off | **Off**<br>**"INIT":** The device is in the INIT state. |
| | Green | **On**<br>**"OPERATIONAL"** |
| | Green | **Blink**<br>**"PRE-OPERATIONAL"** |
| | Green | **Single flash**<br>**"SAFE-OPERATIONAL"** |
| **ECAT-OUT** | | |
| LINK | Off | No EtherCAT device connected, no connection |
| | Green | **On**<br>EtherCAT device is connected |
| | Green | **Blink**<br>The device sends/receives Ethernet frames |
| TX/RX | Off | This LED is not used |

Tab. 11.2: Information about the light behavior of the EtherCAT status LEDs

| LED states | Description |
|---|---|
| On | The indicator is constantly on. |
| Off | The indicator does not come on. |
| Blink | The indicator is switched on or off in phases at a frequency of 2.5 Hz. |
| Single flash | The indicator shows one short flash (200 ms) followed by a longer off phase (1000 ms). |
| Double flash | The indicator shows a sequence of two short flashes (200 ms each) interrupted by a short off phase (200 ms).<br>The sequence is finished by a long off phase (1000 ms). |

## 11.2 EtherCAT basics

### General information

Field buses have been established in automation engineering for many years. Since on the one hand there is demand for ever higher speeds, but on the other hand the technical limits have already been reached with this technology, new solutions must be sought.

The Ethernet known from the office world, with its available-everywhere 100Mbit/s, is very fast. The type of cabling used there and the rules governing access rights mean that this Ethernet is not real-time capable. This effect has been rectified with EtherCAT.

### EtherCAT

For EtherCAT: EtherCAT is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

EtherCAT stands for Ethernet for Controller and Automation Technology. It was originally developed by Beckhoff Automation GmbH and is now supported and further developed by the EtherCAT Technology Group (ETG). The ETG is the world's largest international users and manufacturers association for industrial Ethernet with around 1450 member firms (as at October 2010).

EtherCAT is an open Ethernet-based fieldbus system that is standardized in the IEC. As an open fieldbus system, EtherCAT satisfies the user profile for the area of industrial real-time systems.

Unlike traditional Ethernet communications, in EtherCAT the I/O data are exchanged at 100MBit/s in full duplex mode, while the telegram passes through the coupler. Since in this way a telegram reaches lots of devices in the transmit and receive direction, EtherCAT has a useful data rate of over 90%.

The EtherCAT protocol, optimized for process data, is transported directly in the Ethernet telegram. In turn, this can consist of several sub-telegrams, each serving one memory area of the process image.

### Transmission medium

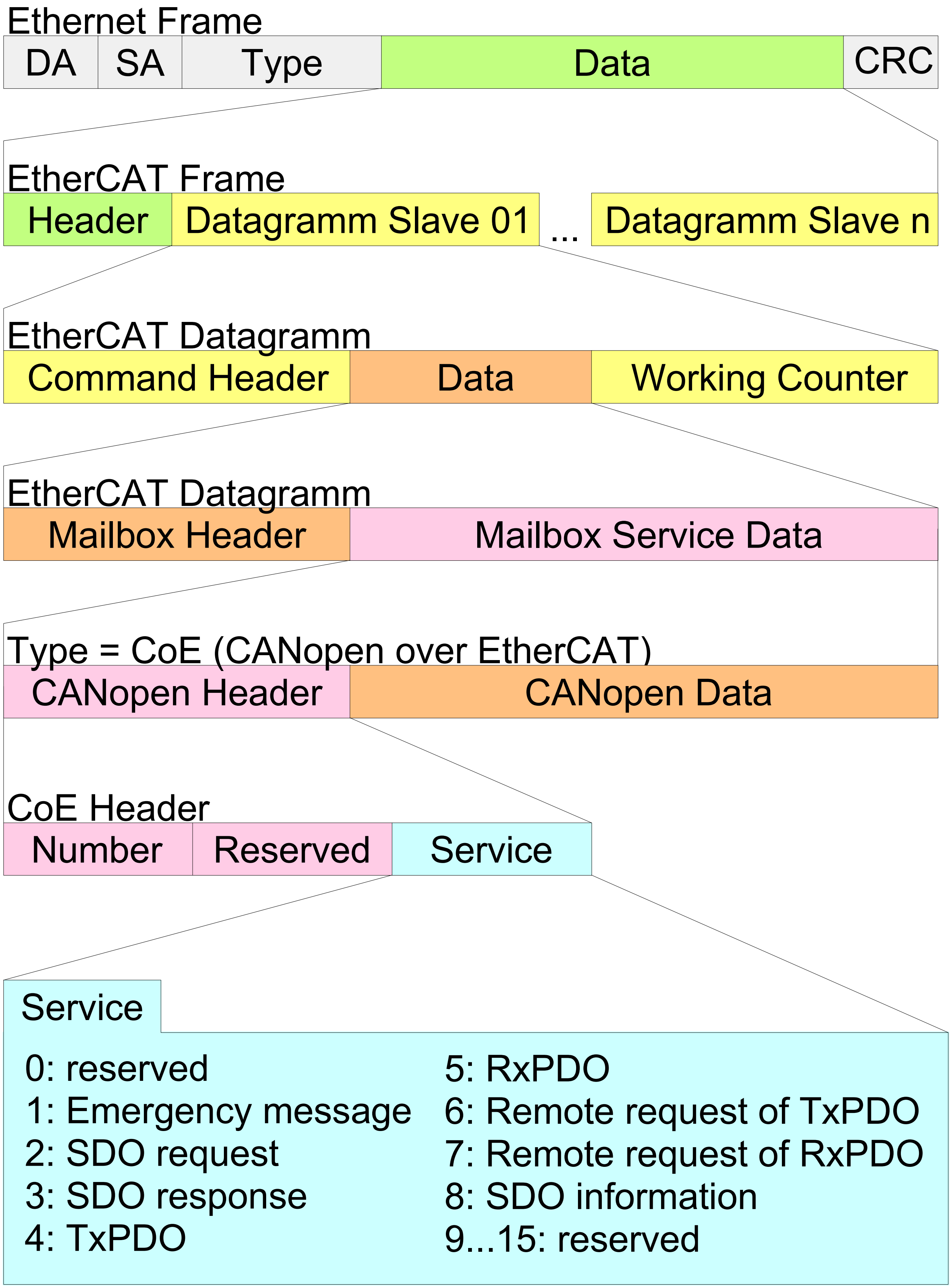
EtherCAT uses Ethernet as the transmission medium. Standard CAT5 cable is used. Cable lengths of up to 100m between 2 devices are possible.

Only EtherCAT components may be used in an EtherCAT network. To implement topologies deviating from the linear structure corresponding EtherCAT components are required that support this. It is not possible to use hubs.

### Communication principle

In EtherCAT the master sends a telegram to the first device. This extracts the data intended for it from the data flow, inserts its response data into the telegram and sends the telegram on to the next device. The next device processes the telegram in the same way.

If the telegram has reached the last device, this recognizes that no more devices are connected and sends the telegram back to the master. In this way the telegram is sent via the other pair of wires through all devices to the master (full duplex). The connection sequence and the use of full-duplex technology means EtherCAT is a logical ring.




| | | | |
|---|---|---|---|
| DA | Destination address | SA | Source address |
| CRC | Checksum | Type | Ether type<br>(example: the entry 0x88A4 means Ether-CAT protocol.) |

### Components

The components of the CoE interface are listed below:

### EtherCAT State Machine

The EtherCAT State Machine controls the state of the EtherCAT coupler.

### Station alias

The EtherCAT address is enumerated automatically by the master. If a special address shall be assigned, the station alias is available. The Leuze EtherCAT slave does not support the allocation of the station alias by the master, but an alias can be set in MSI.designer which is taken over by the slave as an alias if the value is not equal to zero.

**Note:** The transfer of the station alias is only supported from build state A-04. For previous build states, only the automatic negotiation of the address works.

### Object directory

The object directory lists all parameter, diagnostic, process or other data which can be read or described via EtherCAT. The SDO information service provides access to the object directory.

### Process data

The EtherCAT data link layer is optimized for the fast transfer of process data. This determines how the process data of the device is assigned to the EtherCAT process data and how the application on the device is synchronized to the EtherCAT cycle.

The assignment of the process data (mapping) is done via the PDO Mapping and the SyncManager PDO Assign objects. These describe which objects from the object directory are transferred as process data with EtherCAT. The SyncManager Communication objects determine the cycle time with which the associated process data are transferred via EtherCAT and in what form it is synchronized for transmission.

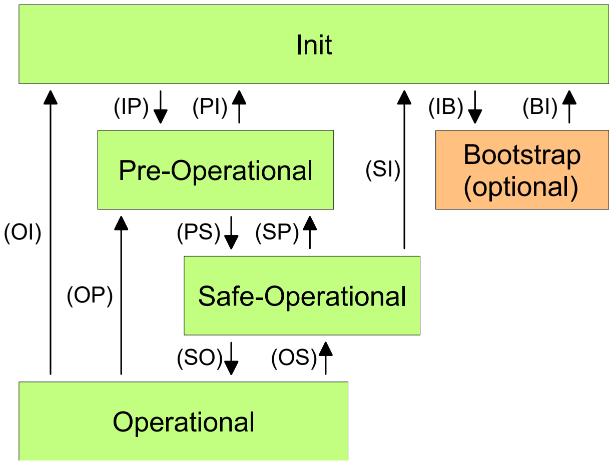
### ESI file: LEUZE_MSI FB ETHERCAT V1.2.xml

You will receive an ESI file from Leuze for the EtherCAT gateway. This file is located either on the enclosed disk or in the download area of www.leuze.com. Install the ESI files in your PLC software configuration tool. Further details on installation of the ESI files can be found in the PLC manual.

## 11.3 EtherCAT state machine

### States

A state machine is implemented in every EtherCAT coupler. For each state it is defined which communication services are active via EtherCAT. The state machine is controlled by the EtherCAT master.




| | | | |
|---|---|---|---|
| IP | Start mailbox communication | PI | Stop mailbox communication |
| PS | Start input update | SP | Stop input update |
| SO | Start output update | OS | Stop output update |
| OP | Stop input update, stop output update | SI | Stop input update, stop mailbox communication |
| OI | Stop output update, stop input update<br>Stop mailbox communication | IB | Start mailbox for firmware update in bootstrap mode (not implemented) |
| BI | Restart/stop mailbox | | |

### INIT

After being switched on, the EtherCAT coupler is in the "Init" state. In this state neither mailbox nor process data communication are possible. The EtherCAT master initializes the SyncManager channels 0 and 1 for mailbox communication.

### Pre-Operational (Pre-OP)

In the transition from **Init** to **Pre-Op**, the EtherCAT checks whether the mailbox was initialized correctly.

In the **Pre-Op** state mailbox communication is possible but not process data communication. Furthermore, in this state the settings for the transmission of process data and module-specific parameters are transmitted where they deviate from the standard settings.

### Safe-Operational (Safe-OP)

In the transition from Pre-Op to Safe-Op the EtherCAT coupler checks whether the channels for process data communication are correct. Before it acknowledges the state change, the EtherCAT gateway copies current output data into the corresponding DP RAM areas of the EtherCAT gateway controller. In the Safe-Op state mailbox and process data communication are possible. Here the output data are updated cyclically while the input data are set to zero.

### Operational (Op)

In the "Op" state the EtherCAT gateway copies the data in the RX-PDO onto its input data set 1. The output data set 1 is copied by the gateway into the TX-PDO and sent to the EtherCAT master.

### Bootstrap optional (Boot)

not implemented

## 11.4 Bus topology and cabling

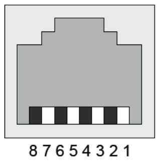
EtherCAT uses Ethernet as the transmission medium. Standard CAT5 cable is used. Cable lengths of up to 100m between 2 devices are possible.

Only EtherCAT components may be used in an EtherCAT network. To implement topologies deviating from the linear structure corresponding EtherCAT components are required that support this. It is not possible to use hubs.

An EtherCAT network always consists of a master and any number of EtherCAT slaves (gateways or couplers). Each EtherCAT slave has an RJ45 socket **IN** and **OUT**. The incoming EtherCAT cable from the direction of the master should be plugged into the socket labeled **IN**. The RJ45 socket **ECAT-OUT** is used to connect further EtherCAT devices in the same strand in order to create so-called "daisy chains". In the last device the **OUT** socket remains free.

### EtherCAT RJ45 bus interface

**Note:** The device supports the Auto Crossover function.

| Pin | Signal | Design |
|---|---|---|
| 1 | TX+ |  |
| 2 | TX- | |
| 3 | RX+ | |
| 4 | Term 1 | |
| 5 | Term 1 | |
| 6 | RX- | |
| 7 | Term 2 | |
| 8 | Term 2 | |
| Housing | Screen | |

Tab. 11.3: Ethernet connection data

| Pin | Signal |
|---|---|
| Medium | 2 x 2 pair twisted copper cable, CAT5 (100 MBit/s) |
| Cable length | max. 100m |
| Transfer rate | 100 MBit/s |

### Important notes

- Use of hubs:
  Hubs are generally **not permitted** in EtherCAT networks.
- Use of switches:
  Switches in EtherCAT networks are only permitted between EtherCAT master and the first EtherCAT slave (100 Mbit/s, full duplex). Leuze electronic GmbH offers its own switches under the product family name "Ethernet Switch".
- Terminator:
  If the gateway is the last device, the EtherCAT topology does not require a terminator.
- Recommendation

  Take appropriate measures to protect the data cables and connectors against high mechanical load. We recommend a fixed installation in conjunction with tension relief.

## 11.5 Data transferred into the network

**Available data**

The MSI 400 EtherCAT gateway can provide the following data:

- Process data
  - **Logic results from the** MSI 400 (see *Routing Table [chapter 5.1.3]*)
  - **Input values** (HIGH/LOW) for all MSI 400 input expansion modules in the system
  - **Output values** (HIGH/LOW) for all MSI 400 input/output expansion modules (see *Module state / input and output values [chapter 3.3.1]*)
  - **Output data** from another network, i.e. data received from a second gateway in the MSI 400 system (see *Transmission of data from a second network [chapter 3.3.3]*)
- Diagnostics
  - **Test values** (CRCs) (see *Data set 2 [chapter 11.5.2]*)
  - **Error and state information** for all modules (see *Error and state information for modules [chapter 3.3.4]*)

**Data sets**

The physical MSI 400 modules are not presented as typical hardware modules in the network. Instead, the data provided by the MSI 400 system have been arranged in three input data sets.

### 11.5.1 Data set 1

Data set 1 (50 bytes) contains the process data. It can be compiled with the aid of MSI.designer. In the form in which it is delivered, the content of data set 1 is preconfigured; it can be freely modified.

Note: **Not allocated** means that the byte value is equal to 0x00. However, the user can freely assign these bytes.

| | *NOTICE* |
|---|---|
| ! | A minimum of one byte must be defined in one of the output data blocks and one byte in one of the input data blocks of dataset 1. |

Tab. 11.4: Data set: Output data set 1 MSI 400 to --> MSI-FB-ETHERCAT

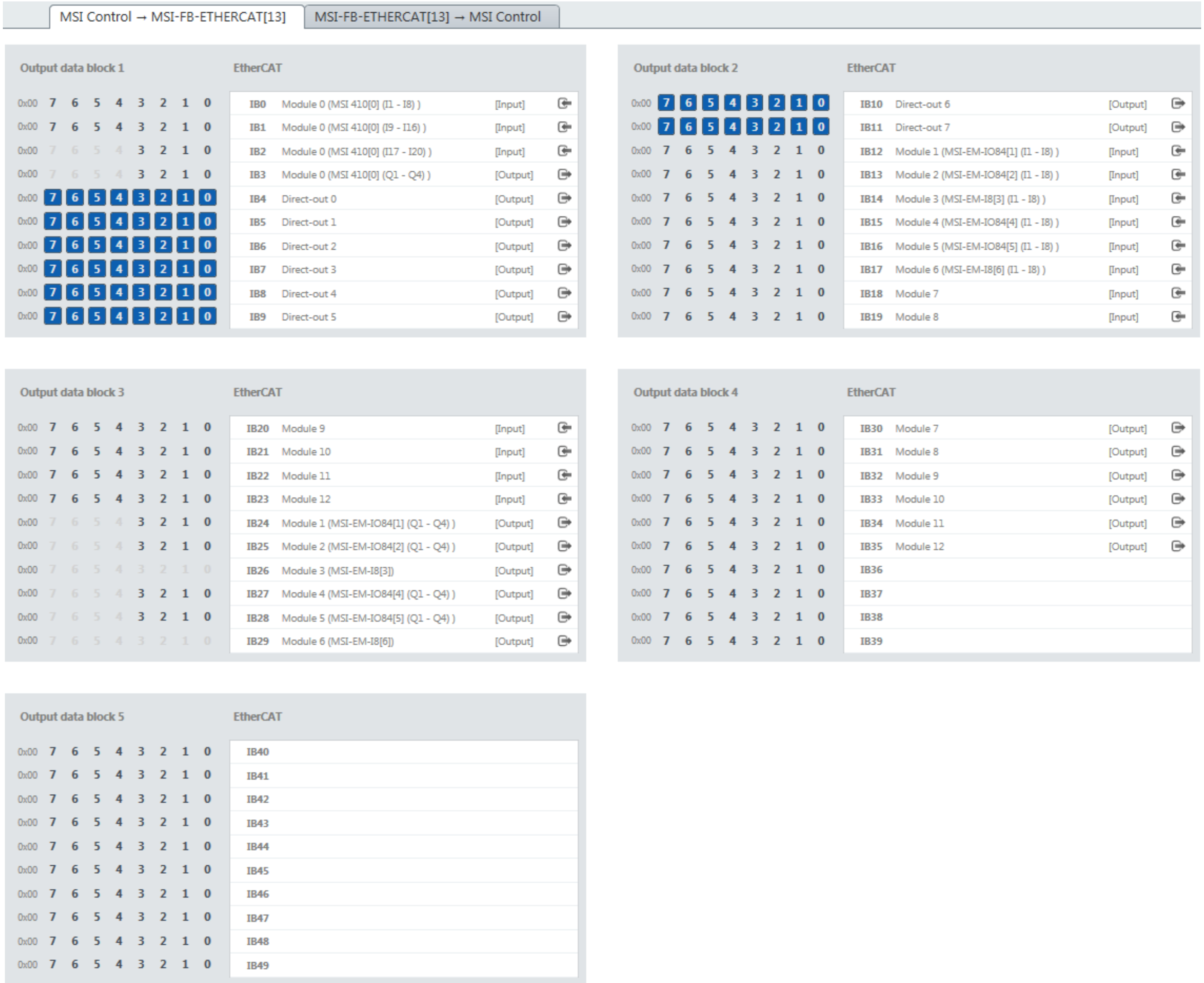
| **Output data block 1** | | **Output data block 2** | |
|---|---|---|---|
| Byte 0 | Input values for Module 0 (I1..I8) | Byte 10 | Not allocated |
| Byte 1 | Input values for Module 0 (I9..I16) | Byte 11 | Not allocated |
| Byte 2 | Input values for Module 0 (IQ1..IQ4) | Byte 12 | Input values for Module 1 |
| Byte 3 | Output values for Module 0 (Q1..Q4, IQ1..IQ4) | Byte 13 | Input values for Module 2 |
| Byte 4 | Not allocated | Byte 14 | Input values for Module 3 |
| Byte 5 | Not allocated | Byte 15 | Input values for Module 4 |
| Byte 6 | Not allocated | Byte 16 | Input values for Module 5 |
| Byte 7 | Not allocated | Byte 17 | Input values for Module 6 |
| Byte 8 | Not allocated | Byte 18 | Input values for Module 7 |
| Byte 9 | Not allocated | Byte 19 | Input values for Module 8 |
| **Output data block 3** | | **Output data block 4** | |
| Byte 20 | Input values for Module 9 | Byte 30 | Output values for Module 7 |
| Byte 21 | Input values for Module 10 | Byte 31 | Output values for Module 8 |
| Byte 22 | Input values for Module 11 | Byte 32 | Output values for Module 9 |

| Output data block 3 | | Output data block 4 | |
|---|---|---|---|
| Byte 23 | Input values for Module 12 | Byte 33 | Output values for Module 10 |
| Byte 24 | Output values for Module 1 | Byte 34 | Output values for Module 11 |
| Byte 25 | Output values for Module 2 | Byte 35 | Output values for Module 12 |
| Byte 26 | Output values for Module 3 | Byte 36 | Not allocated |
| Byte 27 | Output values for Module 4 | Byte 37 | Not allocated |
| Byte 28 | Output values for Module 5 | Byte 38 | Not allocated |
| Byte 29 | Output values for Module 6 | Byte 39 | Not allocated |

| Output data block 5 | |
|---|---|
| Byte 40 | Not allocated |
| Byte 41 | Not allocated |
| Byte 42 | Not allocated |
| Byte 43 | Not allocated |
| Byte 44 | Not allocated |
| Byte 45 | Not allocated |
| Byte 46 | Not allocated |
| Byte 47 | Not allocated |
| Byte 48 | Not allocated |
| Byte 49 | Not allocated |
| **Total length** | **50 bytes** |

**Tag names pre-assigned in the software for the EtherCAT gateway**

The data set 1 is divided into five input data blocks for clarity, whereby data blocks 1 to 5 each contain 10 bytes.




### Direct gateway output values

It is possible to write values directly from the logic editor to the gateway. These valuesare freely programmable and are transferred to the EtherCAT network in the Transmit PDO. Four bytes have been reserved for this purpose in the basic settings for data set 1; however, up to the total number of 50 bytes of data set 1 may be configured as direct gateway output values. Please see the following for more information: *Direct gateway output values [chapter 3.3.1]*

### Module state / input and output values

The MSI 400 gateway can transmit the input and output states of all MSI 400 modules connected to the MSI 400 system over to the network. Data set 3 contains a non-modifiable configuration. Moreover, data set 1 can be adapted to contain up to 4 bytes of collective state information. Only the input and output values for data set 1 have been predefined and these can be freely adapted. You will find more detailed information in the section on the relevant gateway, as well as in the following section: *Configuration of gateways with MSI.designer [chapter 5]*

### Module state

The MSI 400 gateway can transfer the state of the linked modules to the network. A total of 4 bytes are available for this purpose.

Tab. 11.5: Module state

| Module state | Size | Meaning | Assignment |
|---|---|---|---|
| **Input data state** | 2 bytes | One sum bit per module for the state of the module inputs<br>0 = error 1 = no error | Bit 0 = MSI 4xx<br>Bit 1 = 1st module<br>Bit 2 = 2nd module<br>…<br>Bit 12 = 12th module<br>Bit 13 = 1st gateway<br>Bit 14 = 2nd gateway<br>Bit 15 = reserved |
| **Output data state** | 2 bytes | One sum bit per module for the state of the module outputs<br>0 = error 1 = no error | |

You can find information about the meaning of the status bits here in the software manual in chapter "Internal inputs for controller modules"

- **Input values for I/O modules**

  1 byte for data set 1 is available for every expansion module. The input values show the state of the preliminary evaluation of the I/O module. This corresponds to the state of the element in the controller module logic. The level at the associated terminal cannot be clearly detected from this, as the data may be set to low, irrespectively of the level at the input terminal, by means of the cross-connection detection or two-channel evaluation (e.g. I1-18).

  When two-channel input elements have been configured for an I/O module, only the lower-value bit represents the pre-evaluation state of the corresponding element (e.g. bit 0 for I1 and I2, bit 2 for I3 and I4, bit 4 for I5 and I6, bit 6 for I7 and I8). The higher-value bit (bit 1, 3, 5 and 7) is used as follows in this case:

  0 = error 1 = no error

Tab. 11.6: Module status (input data status, byte 1)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Module 7 | Module 6 | Module 5 | Module 4 | Module 3 | Module 2 | Module 1 | MSI 4xx |

Tab. 11.7: Module state (input data state, byte 2)

| Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
|---|---|---|---|---|---|---|---|
| Reserved | Gateway 2 | Gateway 1 | Module 12 | Module 11 | Module 10 | Module 9 | Module 8 |

- **Output values for I/O modules**

  1 byte for data set 1 is available for every module with outputs. The output values indicate the state of the control information from the logic of the controller module for the relevant element of the I/O module. The level of the associated terminals cannot be clearly detected from this, as the output may be switched off via the cross-connection detection or the overload connection function.

  When two-channel output elements have been configured for an I/O module, only the lower-value bit represents the control information (e.g. bit 0 for Q1 and Q2, bit 2 for Q3 and Q4, bit 4 for Q5 and Q6, bit 6 for Q7 and Q8). The higher-value bit (bit 1, 3, 5 and 7) is not used as follows in this case (low):

Tab. 11.8: Module status (output data status, byte 1)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Module 7 | Module 6 | Module 5 | Module 4 | Module 3 | Module 2 | Module 1 | MSI 4xx |

Tab. 11.9: Module state (output data state, byte 2)

| Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
|---|---|---|---|---|---|---|---|
| Reserved | Gateway 2 | Gateway 1 | Module 12 | Module 11 | Module 10 | Module 9 | Module 8 |

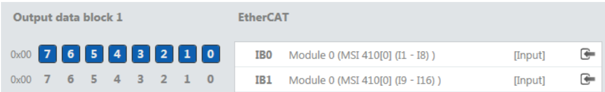
### Transmission of data from a second network

If your MSI 400 system contains two gateways, it is possible to forward information which the first gateway receives from a network (e.g. from an EhterCAT PLC) via the second gateway to a second network (e.g. to a PROFIBUS master) and vice versa.

### Expert setting: Allocating bytes to other addresses

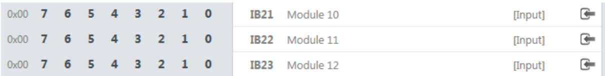
MSI.designer has pre-assigned the addresses according to a default. You can manually change this address allocation by moving any number of bytes.

In our example, we have shifted **byte 1** to **byte 23** in output data block **0**.

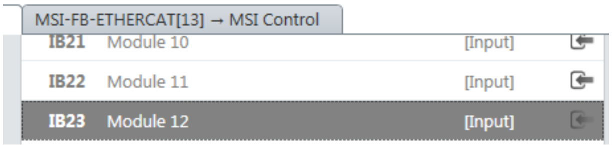



### Step 1: Check target address

- Ensure that the desired address (**byte 23** in our example) has not been allocated.

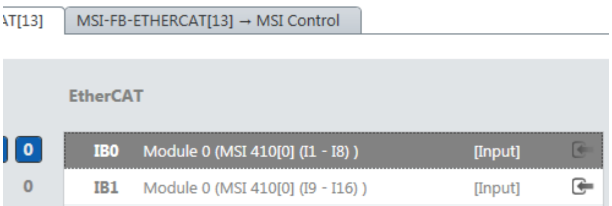



- When the target address is assigned here, delete the bytes placed there.
  To do this, click on the byte in the work area and click on the **Delete** symbol in the command bar.



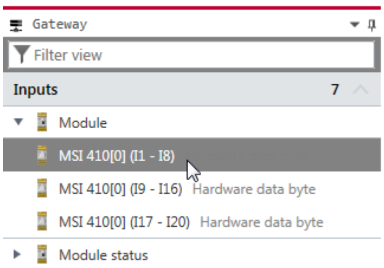

### Step 2: Delete byte from original address

- Delete the byte you wish to reallocate (**byte 0** in our example).
  To do this, click on the byte in the work area and click on the **Delete** symbol in the command bar.

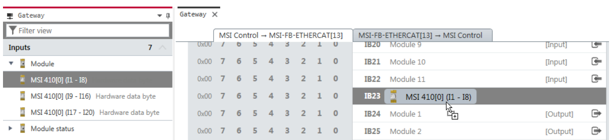



### Step 3: Place byte on new target address

- Open the **Gateway** docking window and select the desired bytes under the associated module.




- Use the mouse button to drag the Byte into the work area on **byte 23**.



### 11.5.2 Data set 2

Data set 2 (32 bytes) contains the test values (CRCs) for the system configuration.

**Configuration check values (CRCs)**

Data set 2 contains the following configuration check values of the MSI 400 system: Project CRC of the project file created with MSI.designer.

The CRC is 4 bytes long. Data set 2 can be read only. The data (Project CRC, System CRC) is available in Little Endian format.

For Modbus/TCP, the project CRC is transmitted in Big Endian format but is transmitted in Little Endian format in the other gateways.

Tab. 11.10: Output data set 2 MSI 400 to --> MSI-FB-ETHERCAT

| Byte | Assignment |
|---|---|
| Byte 0 | Project CRC<br>Value is on the first page in the project report from MSI.designer.<br>Example: CRC Station 1: 0x2ac78506 |
| Byte 1 | |
| Byte 2 | |
| Byte 3 | |
| Byte 4 | Internal CRC[1)] |
| Byte 5 | |
| Byte 6 | |
| Byte 7 | |
| Byte 8 to byte 31 | Reserved for the future |
| **Length** | **32 bytes** |

[1)] The use of the internal CRC in dataset 2 is only permitted for diagnostic purposes so that Leuze Technical Support can continue to provide support.

### 11.5.3 Data set 3

Data set 3 (60 bytes) contains the state and diagnostic data for the various modules, with four (4) bytes per module, with the controller module comprising 3 x 4 bytes. For more details, see Table *"Meaning of module state bits of the secure I/O modules" [chapter 3.3.4]*.

**Error and state information for the modules**

Data set 3 contains the state information for the modules that will be transferred to the network.

Ten bytes are transmitted for each controller module. For each MSI-EM-I8 or MSI-EM-IO84 I/O module, four bytes are transmitted in the Little Endian format, e.g. as a 32-bit word, with the first byte being placed into the least significant byte of the whole number (extreme left) and the fourth byte into the most significant byte of the whole number (extreme right).

Data set 3 cannot be changed.

**NOTICE**

- Reserved (for future use) = static 1 (no state change)
- Not used (can be 0 or 1 ), both values occur.
- If there is no module, all values - including the reserved values - are set to logical 1.

Tab. 11.11: Output data set 3 MSI 400 to --> MSI-FB-ETHERCAT

| Byte | Assignment |
|---|---|
| Byte 0 | Module state MSI 4xx |
| Byte 1 | Module state MSI 4xx |
| Byte 2 | Test impulse comparison MSI 4xx inputs |
| Byte 3 | Test impulse comparison MSI 4xx inputs |
| Byte 4 | Test impulse comparison MSI 4xx inputs |
| Byte 5 | State of dual-channel MSI 4xx inputs |
| Byte 6 | State of dual-channel MSI 4xx inputs |
| Byte 7 | Reserved |
| Byte 8 | Stuck-at error at MSI 4xx outputs |
| Byte 9 | Stuck-at error at MSI 4xx outputs |
| Byte 10 | Reserved |
| Byte 11 | Reserved |
| Byte 12 | Status of Module 1 |
| Byte 13 | Status of Module 1 |
| Byte 14 | Status of Module 1 |
| Byte 15 | Status of Module 1 |
| Byte 16 | Status of Module 2 |
| Byte 17 | Status of Module 2 |
| Byte 18 | Status of Module 2 |
| Byte 19 | Status of Module 2 |
| Byte 20 | Status of Module 3 |
| Byte 21 | Status of Module 3 |
| Byte 22 | Status of Module 3 |
| Byte 23 | Status of Module 3 |
| Byte 24 | Status of Module 4 |
| Byte 25 | Status of Module 4 |
| Byte 26 | Status of Module 4 |
| Byte 27 | Status of Module 4 |
| Byte 28 | Status of Module 5 |
| Byte 29 | Status of Module 5 |
| Byte 30 | Status of Module 5 |
| Byte 31 | Status of Module 5 |
| Byte 32 | Status of Module 6 |
| Byte 33 | Status of Module 6 |
| Byte 34 | Status of Module 6 |
| Byte 35 | Status of Module 6 |
| Byte 36 | Status of Module 7 |
| Byte 37 | Status of Module 7 |
| Byte 38 | Status of Module 7 |

| Byte | Assignment |
|---|---|
| Byte 39 | Status of Module 7 |
| Byte 40 | Status of Module 8 |
| Byte 41 | Status of Module 8 |
| Byte 42 | Status of Module 8 |
| Byte 43 | Status of Module 8 |
| Byte 44 | Status of Module 9 |
| Byte 45 | Status of Module 9 |
| Byte 46 | Status of Module 9 |
| Byte 47 | Status of Module 9 |
| Byte 48 | Status of Module 10 |
| Byte 49 | Status of Module 10 |
| Byte 50 | Status of Module 10 |
| Byte 51 | Status of Module 10 |
| Byte 52 | Status of Module 11 |
| Byte 53 | Status of Module 11 |
| Byte 54 | Status of Module 11 |
| Byte 55 | Status of Module 11 |
| Byte 56 | Status of Module 12 |
| Byte 57 | Status of Module 12 |
| Byte 58 | Status of Module 12 |
| Byte 59 | Status of Module 12 |
| **Length** | **60 bytes** |

## 11.6 Data received from the network

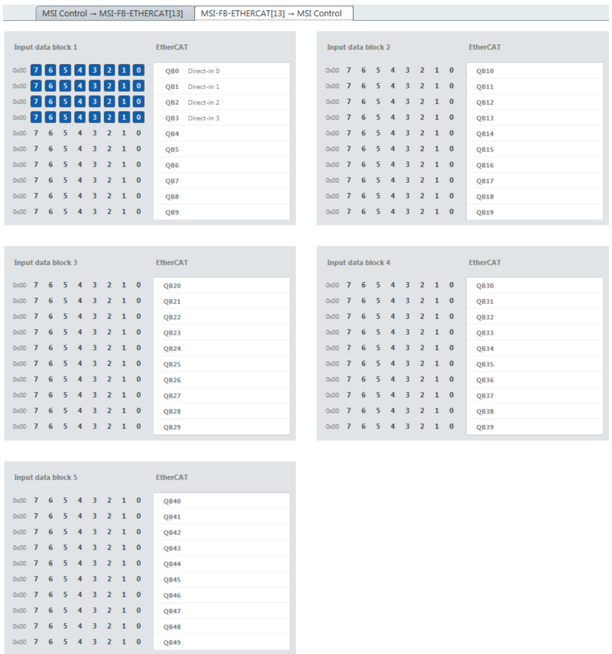
The data received from the network is divided into five data blocks of 10 bytes each for clarity.

The content of the input data blocks can be used in the logic editor of the MSI.designer, as well as made available for another network via a second gateway within the MSI 400 system.

***NOTICE***

- In order to use network data in the logic editor or as input for another network, you must assign a tag name for each bit to be used.
- Bits without specific tag names will not be available in the logic editor or for routing via a second gateway. Detailed information about how to assign tag names for the data received may be found in the corresponding sections of the chapters on the various gateways.
- You can monitor current communication with the network with the aid of input data state bits for receiving data from the network and the output data state bit for transmitting data to the network in the logic editor. When the gateway detects a communication error, both the content of the data sets and the associated state bit are set to zero (logical 0).
- When all communication fails, the data of the output data sets and the input data state bit are set to zero (logical 0).
- When a connection is closed while others remain available, the LED MS or LED state will flash red/green for a total of 10 seconds and an entry will be made in the error log. In this case the state bits are not affected.

### Tag names pre-assigned in the software for the EtherCAT gateway



### Delete any bytes not required

You can delete bytes pre-allocated by MSI.designer that you do not require by clicking on them with the mouse.

- Launch MSI.designer.
- Read the hardware configuration, including the MSI-FB-ETHERCAT gateway.
  Instructions: Software manual, chapter "Connecting to the safety controller"
- Switch to the **Gateway** view.
- Click on the byte you do not need and wish to delete.


- Click on the **Delete** icon in the command bar.

You will find further information about how to configure the process diagram here:

- *Configuration of gateways with MSI.designer [chapter 5]*
- Software manual

### Structure of the data block

The input data block consists of 50 bytes (byte 0 to 49) of data that is transferred from the EtherCAT network to the MSI-FB-ETHERCAT gateway. The content of the data bytes does not meet the requirements of a safety system. The values are only current as long as the gateway to the EtherCAT network is connected and the gateway status is **Operational**. As soon as the state machine of the gateway adopts a state other than **Operational**, this data is set to zero.

Also see: *Gateway state machine [chapter 11.3]*

Tab. 11.12: Input data block 1–5 of the MSI-FB-ETHERCAT module to --> MSI 400

| | Input data block 1 | Input data block 2 | Input data block 3 | Input data block 4 | Input data block 5 |
|---|---|---|---|---|---|
| **Byte 0** | Byte 0 | Byte 10 | Byte 20 | Byte 30 | Byte 40 |
| **Byte 1** | | | | | |
| **Byte 2** | | | | | |
| **Byte 3** | | | | | |
| **Byte 4** | | | | | |
| **Byte 5** | | | | | |
| **Byte 6** | | | | | |
| **Byte 7** | | | | | |
| **Byte 8** | | | | | |
| **Byte 9** | Byte 9 | Byte 19 | Byte 29 | Byte 39 | Byte 49 |
| **Length** | 10 bytes | 10 bytes | 10 bytes | 10 bytes | 10 bytes |

## 11.7 Configuring an EtherCAT network

A device description file (ESI = **E**therCAT **S**lave **I**nformation) in XML format is delivered with the MSI-FB-ETHERCAT. The EtherCAT master integrates this file into the EtherCAT system so that the master has the necessary EtherCAT configuration data and can establish a connection to the gateway.

Please read the manual depending on your controller to see which steps are required in detail.

## 11.8 EtherCAT configuration of the gateway - how the data are transferred

The following steps are required to configure the communication between the PLC programming system and the gateway. Configuration in the programming system is done by integrating a standardized ESI description file.

***NOTICE***

This documentation does not address the installation of the EtherCAT network or the other components of the automation system project in the network configuration tool. It is assumed that the EtherCAT project in the configuration program (e.g. Beckhoff TwinCAT) has already been set up. The examples presented are based on configurations created with the help of Beckhoff TwinCAT.

### Step 1: Install the EtherCAT slave description file

Before the MSI-FB-ETHERCAT module can be used for the first time as part of the network configuration tool (e.g. Beckhoff TwinCAT), the gateway description file must first be installed in the hardware catalog of the tool.

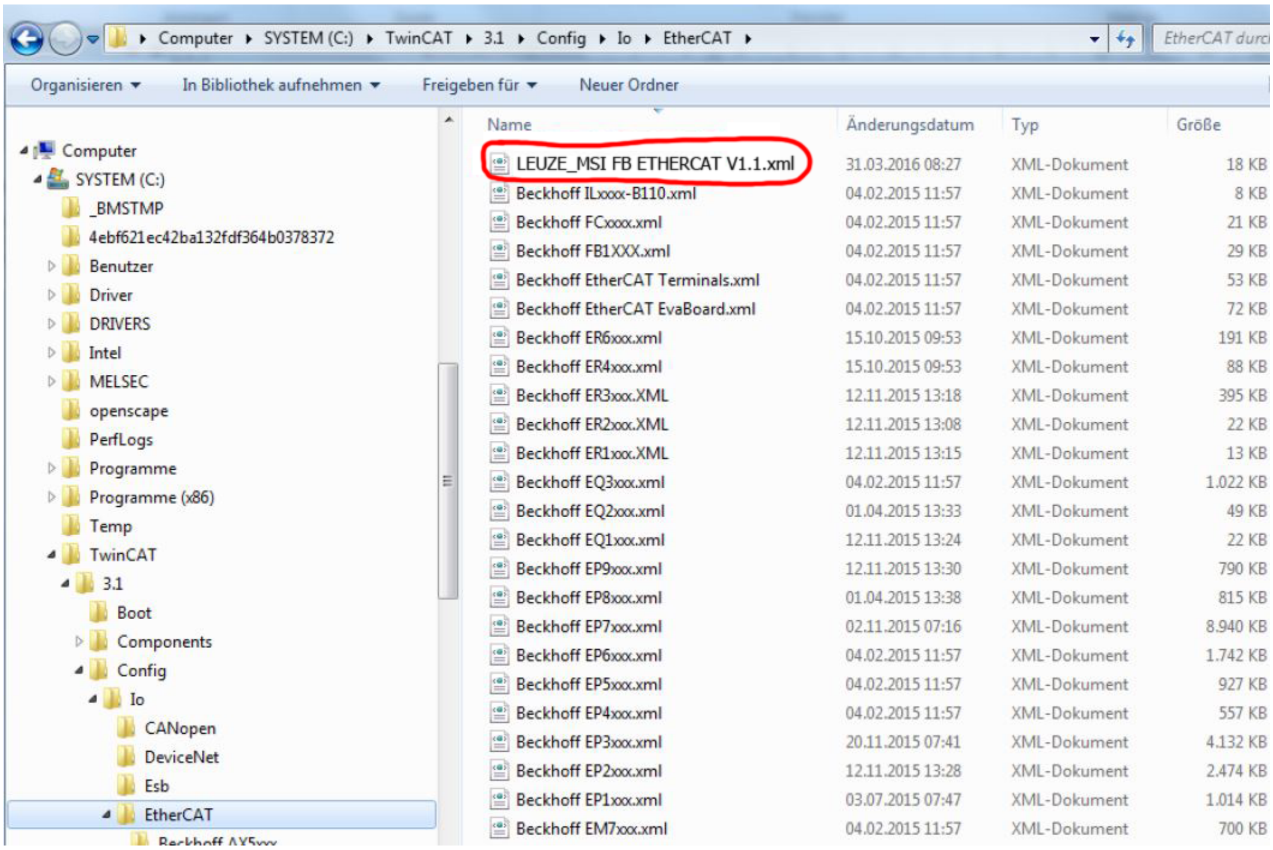
- ✎ Download the GSD file and the equipment symbol from the product site of the MSI-FB-ETHERCAT module (www.leuze-shop.com).
- ✎ Follow the instructions to install XML in the online help or user manual of the EtherCAT network configuration tool for the master or for the EtherCAT control system.

### Step 2: Add the gateway to a PLC project

To make the system of the MSI 400 system available in the process diagram of the PLC, the gateway must first be added to the hardware configuration. The procedure to be used depends on the hardware configuration software of the PLC used. Please also read the documentation for the corresponding software in this regard.

The example below shows how the gateway is added to a control project in Beckhoff TwinCAT.

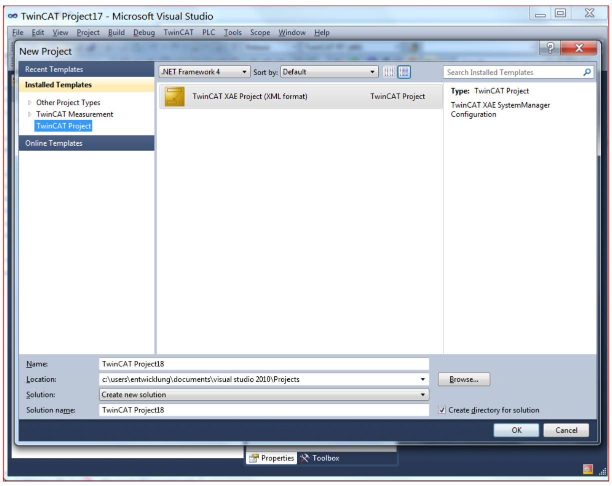
- ✎ Copy the description file LEUZE_MSI FB ETHERCAT V1.1.xml to the TwinCAT folder. An example of a typical installation can be seen below:

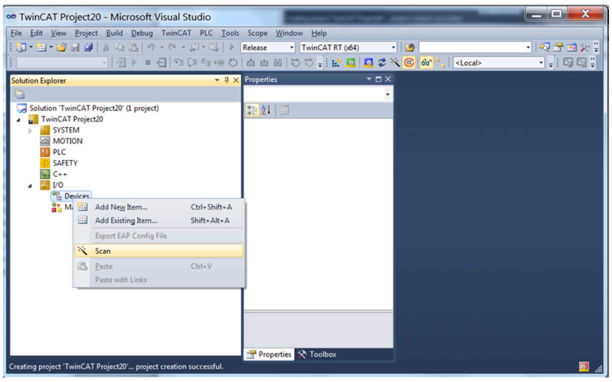



- If a path is to be specified in the ESI file in which, for example, the description file for the expansion modules is locates, create this path in the directory exactly as described in the file.
- Re-start TwinCAT.
  Note: The folder with the current description files is only read when the program is restarted.

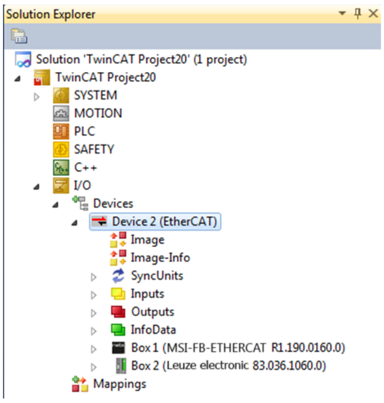
Example: This is not true for the gateway, but is important for other slaves.


## Step 3: Create a new project





After you have connected the controller – i.e. the EtherCAT master to the EtherCAT slaves – you can scan the connected slaves.




TwinCAT shows the found slaves in the Solution Explorer as a box with the corresponding device names.

Tab. 11.13: Error

| Error | Cause |
|---|---|
| Box is displayed with no device names. | ESI file was not found. |
| No EtherCAT slave (box) is displayed. | Modules are not connected to the EtherCAT master or are not powered. |
| The input data (inputs) are not up to date or all have the value 0. | The controller module is at stop or no data was mapped to output data set 1.<br>Data sets 2 and 3 are displayed once the controller module is in the RUN state. |
| The output data (outputs) is transmitted to the gateway but not displayed in the input data sets. | No tags have been created in the input data set. |

## 11.9 Diagnostic LEDs on the gateway and troubleshooting

You can find information about the diagnostics of the MSI 400 system in the software manual.

Tab. 11.14: Troubleshooting the MSI-FB-ETHERCAT module

| Error | Possible cause | Possible remedy |
|---|---|---|
| Key: ○ LED off / ☀ LED flashes / ● LED lights up | | |
| MSI.designer cannot set up a connection to the controller module. | | • Switch on the power supply.<br>• Check the communication settings in MSI.designer. |
| The MSI-FB-ETHERCAT does not provide any input data.<br>MS LED: Red / green | After switch-on:<br>• EtherCAT not connected. | • Connect RJ45 cable to ECAT-IN. |
| The MSI-FB-ETHERCAT does not provide any input data.<br>MS LED: Red / green<br>LINK (ETHERCAT-IN): Green | After switch-on:<br>• RJ45 is connected to the port, no data on the EtherCAT Net. | • Activate EtherCAT. |
| The MSI-FB-ETHERCAT does not provide any input data.<br>MS LED: Red / green<br>LINK (ETHERCAT-IN): Green | After switch-on:<br>• RJ45 is connected to the port, EtherCAT not active. | • Activate EtherCAT and initialize Gateway.<br>**Init** state |
| Controller errors<br>MS LED: Red<br>RUN: Green<br>ERR: Red | • Incorrect EtherCAT configuration, gateway is addressed with incorrect data.<br>• Gateway is in **Pre-Op** state. | • Check network and device configuration.<br>• Switch power off and back on. |
| LINK: Green | | |

| Error | | Possible cause | Possible remedy |
|---|---|---|---|
| The MSI-FB-ETHERCAT does not provide any input data. | | After switch-on:<br>• Gateway state is **Init**. | • Switch EtherCAT to **Op** state. |
| MS LED | Red / green | | |
| RUN | Off | | |
| LINK | Green | | |
| The MSI-FB-ETHERCAT does not provide any input data. | | After switch-on:<br>• Gateway state is **Pre-Op**. | • Switch EtherCAT to **Op** state. |
| MS LED | Red / green | | |
| RUN | Green | | |
| LINK | Green | | |
| The MSI-FB-ETHERCAT does not provide any input data. | | • Gateway state is **Safe-Op**. | • Switch EtherCAT to **Op** state. |
| MS LED | Red / green | | |
| RUN | Green/flash | | |
| LINK | Green | | |
| The MSI-FB-ETHERCAT does not provide any input data. | | • No EtherCAT data, but there is a bus connection to next EtherCAT slave. | • Re-start EtherCAT master or supply master with power.<br>• Check RJ45 cable.<br>• Repair interruption to the EtherCAT network. |
| MS LED | Red / green | | |
| RUN | Green/flash | | |
| ERR | Red / double flash | | |
| LINK (ETHERCAT-IN) | Green | | |

**Notes on troubleshooting**

- **LINK LEDs**

  Use the state of the LINK LEDs to check whether there is a connection to the Ethernet.
- **Cables**

  Check that the pin assignment of the used cable is correct.
- **Configuration**

  Make sure that the gateway is installed right next to the controller module and that no more than 2 MSI 400 gateways are connected. Also ensure that only a maximum of 12 I/O extended modules are connected next to the gateways.
- **Mechanical strength**

  Check whether the RJ 45 connectors are engaged by gently pulling on the EtherCAT connection cables.
  In case of high mechanical load, secure the RJ45 cable with a tension relief.

# 12 Technical data

## 12.1 Modbus TCP, PROFINET IO and EtherNet/IP gateway

Use the MSI 430-x/???- LEUZE controller module for the Modbus TCP, PROFINET IO and EtherNet/IP functionalities.

You will find the technical data for this module here:
Hardware manual, chapter "Controller module"

## 12.2 EtherCAT gateway

| Interface | Minimal | Typical | Maximum |
|---|---|---|---|
| Field bus | EtherCAT | | |
| Connection technology | RJ45 socket | | |
| Transfer rate | 100 Mbit/s (100 Base-TX) | | |
| Device type | EtherCAT slave | | |
| Data length: Inputs | 50 bytes from EtherCAT to MSI 400 | | |
| Data length: Outputs | 142 bytes (50 + 32 + 60) from MSI 400 to EtherCAT | | |
| Galvanic isolation | Yes - between EtherCAT (RJ45) and system voltage | | |
| Type of insulation | Function insulation | | |
| Field bus | EtherCAT | | |

## 12.3 PROFIBUS DP

| Interface | Minimal | Typical | Maximum |
|---|---|---|---|
| Field bus | PROFIBUS-DP-V0 | | |
| Interface level | RS-485 | | |
| Connection technology | 9-pin D-sub socket | | |
| Slave address (set via rotary switch) | 0 | | 99 |
| Slave address (set in MSI.designer[1)) | 3 | | 125 |
| Baud rate (automatic adaptation) | | | 12 MBaud |
| Baud rate (kbits/s with standard line) | | | Maximum line length |
| 9.6/19.2/93.75 | | | 1200 m |
| 187.5 | | | 1000 m |
| 500 | | | 400 m |
| 1.500 | | | 200 m |
| 12.000 | | | 100 m |
| Line parameters | see *PROFIBUS-DP gateway [chapter 9]* | | |
| Galvanic isolation | Yes - between D-Sub socket and system voltage | | |
| Type of insulation | Function insulation | | |
| [1) To set the slave address via software, the hardware address setting must be "0". | | | |

## 12.4 CANopen gateways

| Interface | Minimal | Typical | Maximum |
|---|---|---|---|
| Field bus | CANopen DS-301 | | |
| Interface level | RS-485 | | |
| Connection technology | 5-pin "open style" socket | | |
| Slave address (set via rotary switch) | 0 | | 99 |
| Slave address (set in MSI.designer [1]) | 1 | | 127 |
| Baud rate (kbits/s with standard line) | | | Maximum line length |
| 125 | | | 500 m |
| 250 | | | 250 m |
| 500 | | | 100 m |
| 800 | | | 40 m |
| 1000 | | | 20 m |
| Line parameters | see *CANopen gateway [chapter 10]* | | |
| Galvanic isolation | Yes - between 5-pin socket and system voltage | | |
| Type of insulation | Function insulation | | |
| [1)]To set the slave address via software, the hardware address setting must be "0". | | | |

## 12.5 Technical data for supply circuit

These technical data apply to all gateway modules.

| Supply circuit (e.g. via internal safety bus) | Typical |
|---|---|
| Supply voltage | 24 V DC +25 % / -30 % |
| Power consumption | Max. 2.4 W |

## 12.6 General technical data

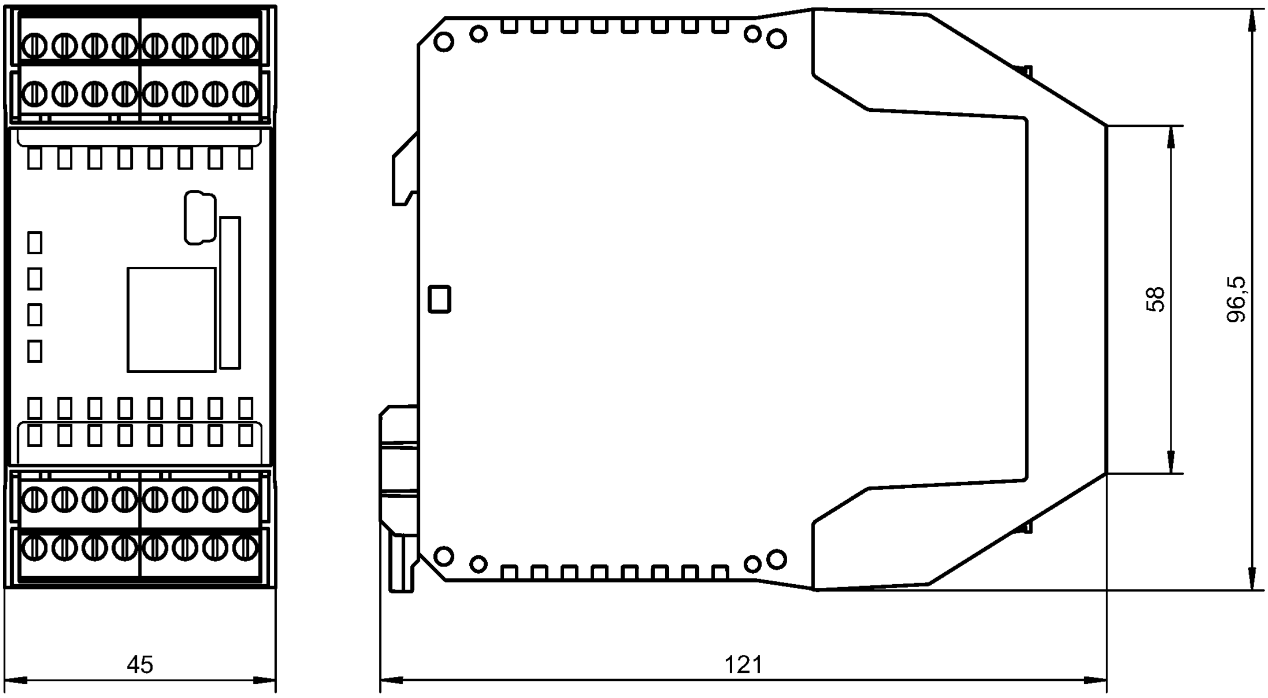
These technical data apply to all gateway modules.

| | General technical data |
|---|---|
| **Connection terminals** | |
| Field bus | See: *Interfaces and operation [chapter 9.1]* |
| **Climatic conditions** | |
| Environmental operating temperature $T_A$ | -25 to +55°C |
| Storage temperature | -25 to +70°C |
| Relative humidity | 10 to 95%, non-condensing |
| Climatic conditions (EN 61131-2)<br>Air pressure during operation | <br>860 to 1060 hPa |
| **Mechanical strength** | |
| Fatigue strength | 5 … 150 Hz (EN 60068-2-6) |
| Shock resistance<br>• Continuous shock<br>• Single shock | <br>10 g, 16 ms (EN 60068-2-29)<br>30 g, 11 ms (EN 60068-2-27) |
| Electric safety | See MSI 4xx |
| Protective type (EN 60529) | IP 20 |
| Protection class | III |
| Electromagnetic compatibility | EN 61000-6-2/EN 55011 Class A |
| **Mechanics and set-up** | |
| Housing material | Polycarbonate |
| Housing type | Device for installation in switch box |
| Housing protection type/terminals | IP 20/IP 40 |
| Color | Light gray |
| Weight | 0.16 kg |
| Internal safety bus | 10-pin plug on the right<br>10-pin bushing on the left |
| DIN rail | DIN rail TH 35 according to EN 60715 |

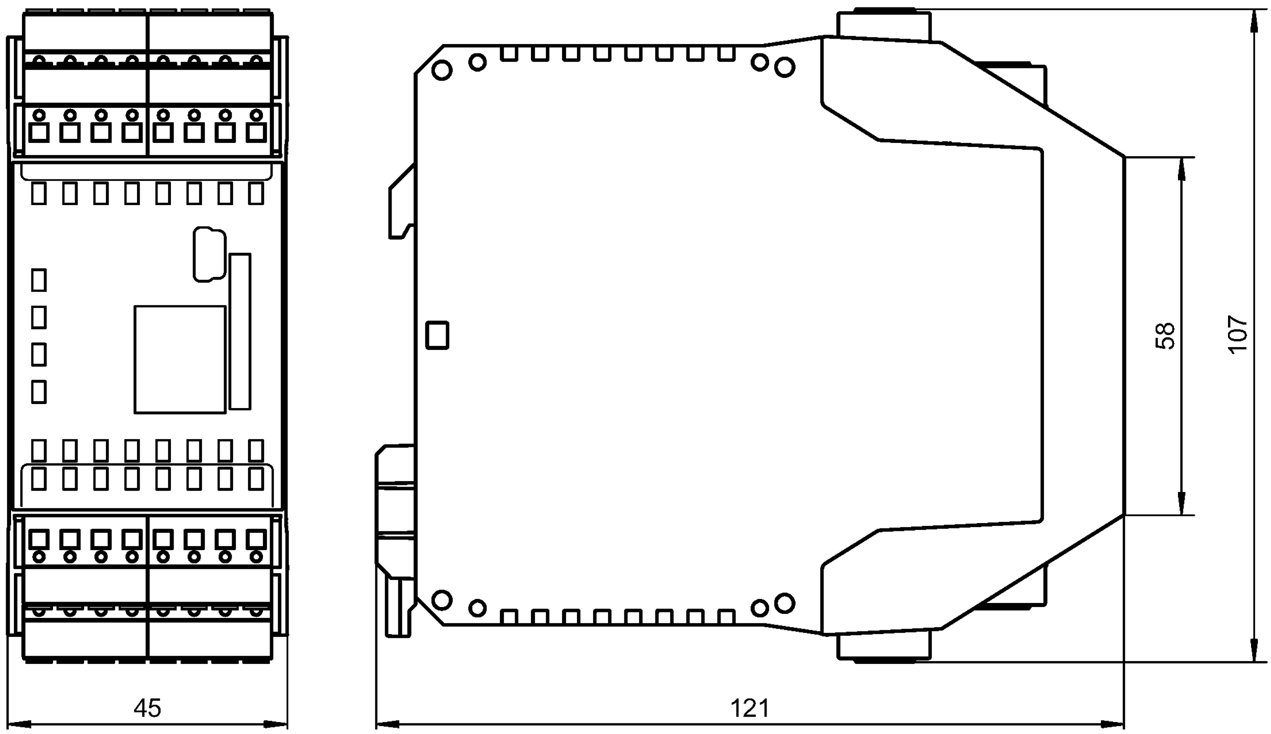
## 12.7 Dimensional drawings

### 12.7.1 Controller module

**Screw terminal**




**Spring-loaded terminal**



### 12.7.2 CANopen and PROFIBUS gateways

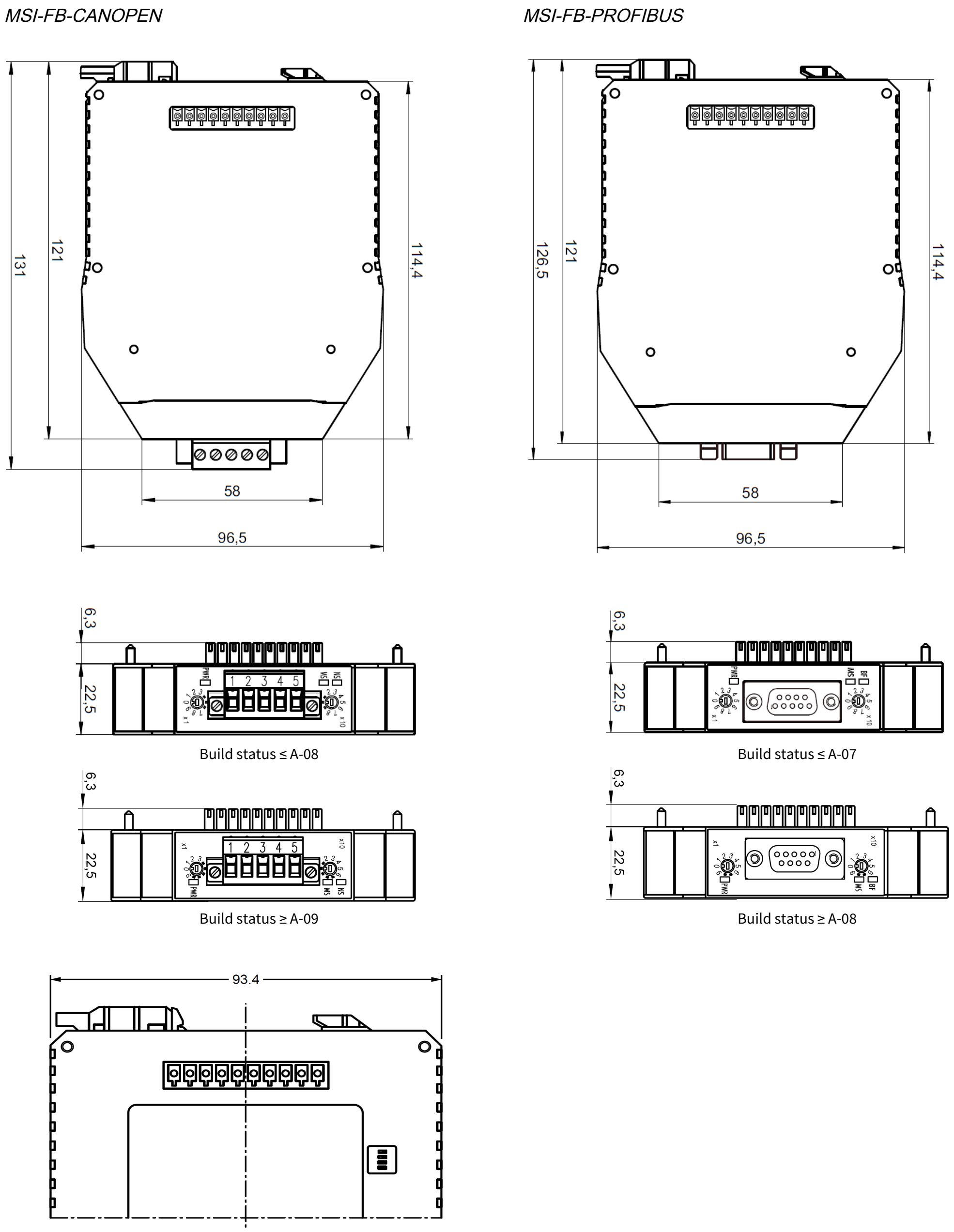



Fig. 12.1: Dimensional drawing CANopen and PROFIBUS gateways (mm)

### 12.7.3 EtherCAT gateway

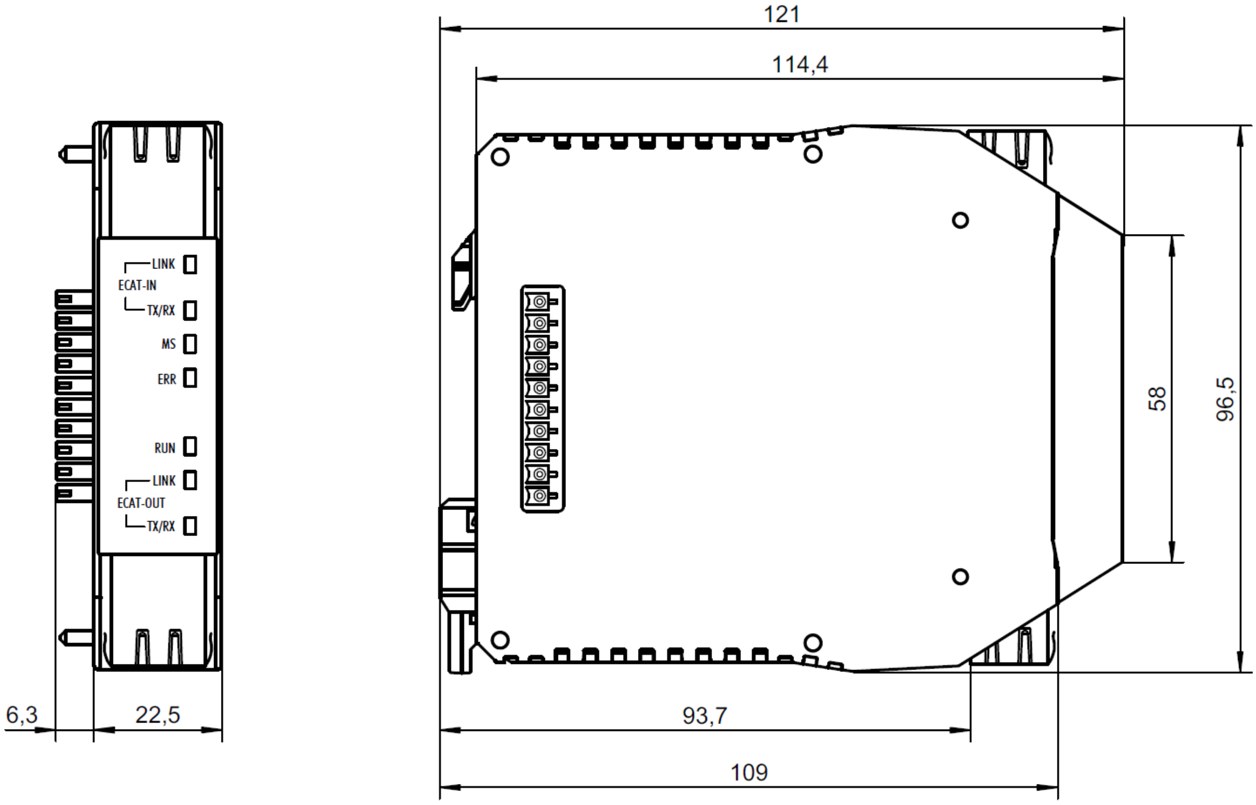



Fig. 12.2: Dimensional drawing EtherCAT gateway (mm)

# 13 Order data

## 13.1 Hardware modules and accessories

Tab. 13.1: Order numbers for the MSI 400 modules

| Type | Description | Part number |
|---|---|---|
| MSI 410-01 | Controller module, USB connection,<br>20 inputs / 4 outputs<br>Screw terminals, pluggable | discontinued |
| MSI 410-03 | Controller module, USB connection,<br>20 inputs / 4 outputs<br>Spring-loaded terminals, pluggable | discontinued |
| MSI 420-01 | Controller module, USB and Ethernet connection,<br>16 inputs / 4 outputs and 4 configurable inputs or out-puts<br>Screw terminals, pluggable | 50132986 |
| MSI 420-03 | Controller module, USB and Ethernet connection,<br>16 inputs / 4 outputs and 4 configurable inputs or out-puts<br>Spring-loaded terminals, pluggable | 50132987 |
| MSI 430-01 | Controller module, USB and industrial Ethernet con-nection,<br>16 inputs / 4 outputs and 4 configurable inputs or out-puts<br>Screw terminals, pluggable | 50132988 |
| MSI 430-03 | Controller module, USB and industrial Ethernet con-nection,<br>16 inputs / 4 outputs and 4 configurable inputs or out-puts<br>Spring-loaded terminals, pluggable | 50132989 |
| MSI-SD-CARD | Program removable storage | 50132996 |
| KB USB A – USB miniB | 1.8 m USB configuration capable | 50117011 |
| MSI-FB-CANOPEN | CANopen gateway | 50132994 |
| MSI-FB-PROFIBUS | PROFIBUS-DP gateway | 50132995 |
| MSI-EM-IO84-01 | Safe input/output expansion with output test pulses<br>8 inputs/4 outputs<br>Screw terminals, pluggable | 50132990 |
| MSI-EM-IO84-03 | Safe input/output expansion with output test pulses<br>8 inputs/4 outputs<br>Spring-loaded terminals, pluggable | 50132991 |
| MSI-EM-I8-01 | Safe input expansion<br>8 inputs<br>Screw terminals, pluggable | 50132992 |
| MSI-EM-I8-03 | Safe input expansion<br>8 inputs<br>Spring-loaded terminals, pluggable | 50132993 |

| Type | Description | Part number |
|---|---|---|
| MSI-EM-IO84NP-01 | Standard input/output expansion<br>4 inputs / 4 outputs and 4 configurable inputs or out-puts<br>Screw terminals, pluggable | 50132997 |
| MSI-EM-IO84NP-03 | Standard input/output expansion<br>4 inputs / 4 outputs and 4 configurable inputs or out-puts<br>Screw-loaded terminals, pluggable | 50132998 |
| MSI-FB-ETHERCAT | EtherCAT Gateway | 50132999 |

## 13.2 Modules for contact expansion

| Type | Description | Part number |
|---|---|---|
| MSI-SR-CM43-01 | Contact expansion, 24 V DC,<br>4 NC contacts, 3 NO contacts,<br>Screw terminals, pluggable | 50133026 |
| MSI-SR-CM43-03 | Contact expansion, 24 V DC,<br>4 NC contacts, 3 NO contacts,<br>Spring-loaded terminals, pluggable | 50133027 |
| MSI-SR-CM42R-01 | Contact expansion with 2 relay groups, 24 V DC,<br>2 x 2 NC (normally closed contact), 2 x 1 NO (normally open contact),<br>Screw terminals, pluggable | 50133014 |
| MSI-SR-CM42R-03 | Contact expansion with 2 relay groups, 24 V DC,<br>2 x 2 NC (normally closed contact), 2 x 1 NO (normally open contact),<br>Spring-loaded terminals, pluggable | 50133015 |