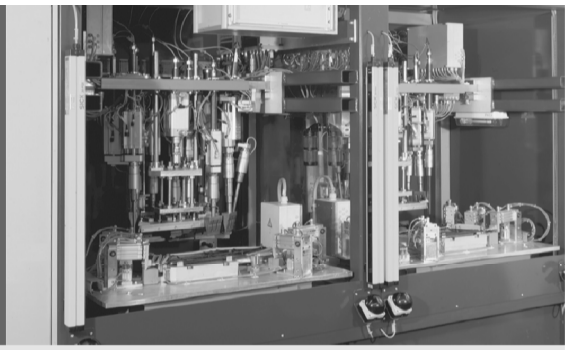
OPERATING INSTRUCTIONS



Flexi Soft

Gateways

GB

SICK
Sensor Intelligence.

Contents

1 About this document 7
1.1 Function of this document 7
1.2 The Flexi Soft operating instructions 7
1.3 Target group 7
1.4 Information depth 8
1.5 Scope 8
1.6 Abbreviations used 8
1.7 Symbols used 9
1.8 Trademarks 9

2 On safety 10
2.1 Qualified safety personnel 10
2.2 Correct use 10
2.3 Environmental protection 11
2.3.1 Disposal 11
2.3.2 Separation of materials 11

3 Product description Flexi Soft gateways 12
3.1 Device variants 13
3.2 Firmware versions 13
3.3 Data transmitted into the network (input data sets) 14
3.3.1 Logic results 17
3.3.2 Gateway direct output values 17
3.3.3 Module status and EFI status as well as input and output values 17
3.3.4 Routing of data from a second network 19
3.3.5 Configuration checksums (CRCs) 19
3.3.6 Error and status information of the modules 19
3.4 Data received from the network (network output data sets) 23

4 Mounting and basic configuration of the gateways 24
4.1 Mounting/Dismantling 24
4.1.1 Steps for mounting the modules 24
4.1.2 Steps for dismantling the modules 26
4.2 Electrical installation 27
4.3 First configuration steps 27
4.3.1 Establishing a connection between gateway and PC 28
4.3.2 Configuration of the gateways 29
4.3.3 Transfer of a configuration 30
4.3.4 Verification of a configuration 30
4.3.5 Upload of a configuration 31

5 Ethernet gateways 32
5.1 Common features of the Ethernet gateways 32
5.1.1 TCP/IP configuration interface 32
5.1.2 Ethernet TCP/IP socket interface 37
5.1.3 TCP/IP process image example 46
5.1.4 TCP/IP socket monitor 48
5.2 EtherNet/IP gateway 52
5.2.1 Interfaces and operation 53
5.2.2 Basic configuration — Assigning a device name and IP address 54
5.2.3 EtherNet/IP Class 1 communication — Implicit messaging 55
5.2.4 Example configuration of implicit messaging with Rockwell RSLogix 5000 60
5.2.5 Example configuration of implicit messaging with an OMRON PLC 60
5.2.6 Ethernet/IP Class 3 communication — Explicit messaging 60
5.2.7 Example for the configuration of explicit messaging 72
5.2.8 TCP/IP configuration interface 73
5.2.9 Ethernet TCP/IP socket interface 73
5.2.10 Diagnostics and troubleshooting 74
5.3 Modbus TCP gateway 75
5.3.1 Interfaces and operation 75
5.3.2 Basic configuration — Assigning an IP address 76
5.3.3 Configuration of the Modbus TCP interface to the PLC — How the data are transferred 77
5.3.4 TCP/IP configuration interface 83
5.3.5 Ethernet TCP/IP socket interface 83
5.3.6 Diagnostics and troubleshooting 83
5.4 PROFINET IO gateway 84
5.4.1 Interfaces and operation 85
5.4.2 Basic configuration — Assigning a device name and IP address 86
5.4.3 PROFINET configuration of the gateway — How the data are transferred 88
5.4.4 PROFINET configuration of the gateway — Which data are transferred 92
5.4.5 TCP/IP configuration interface 99
5.4.6 Ethernet TCP/IP socket interface 99
5.4.7 Diagnostics and troubleshooting 100
5.5 EtherCAT gateway 101
5.5.1 Interfaces and operation 101
5.5.2 Installation of the gateway in the Flexi Soft system 103
5.5.3 EtherCAT configuration of the gateway 104
5.5.4 Input data — Flexi Soft to EtherCAT 107
5.5.5 Output data — EtherCAT to Flexi Soft 107
5.5.6 Tag name export 108
5.5.7 Ethernet over EtherCAT (EoE) 110
5.5.8 TCP/IP configuration interface 110
5.5.9 CoE (CAN application layer over EtherCAT) 111
5.5.10 Diagnostics and troubleshooting 114

Contents

6 Fieldbus gateways 116
6.1 PROFIBUS DP gateway 116
6.1.1 Interfaces and operation 116
6.1.2 Planning 119
6.1.3 PROFIBUS configuration of the gateway — How the data are transferred 122
6.1.4 Diagnostics and troubleshooting 129
6.2 CANopen gateway 130
6.2.1 Interfaces and operation 130
6.2.2 CANopen configuration of the gateway — How the data are transferred 133
6.2.3 CANopen configuration of the gateway — Which data are transferred 136
6.2.4 NMT — Network management 137
6.2.5 SYNC 138
6.2.6 Emergency 138
6.2.7 Node guarding 142
6.2.8 PDO communication 143
6.2.9 SDO communication 145
6.2.10 SDO object directory 146
6.2.11 Guarding protocols 153
6.2.12 Error objects 155
6.2.13 CANopen diagnostics examples 156
6.2.14 Diagnostics and troubleshooting 159
6.3 DeviceNet gateway 160
6.3.1 Characteristic of the DeviceNet implementation 160
6.3.2 Interfaces and operation 161
6.3.3 Setting up DeviceNet communication 164
6.3.4 Supported DeviceNet features 166
6.3.5 DeviceNet protocol settings 167
6.3.6 Assemblies 167
6.3.7 Diagnostics and troubleshooting 171

7 Layout and content of the process image 173
7.1 Routing 173
7.2 Default settings for the operational data 174
7.3 Customizing the operational data (Flexi Soft to Network) 175
7.3.1 The toolbar 176
7.3.2 Available data area 176
7.3.3 Gateway data set to network area 177
7.3.4 Tag names area 177
7.3.5 Gateway direct output values 178
7.3.6 Output data configuration (Network to Flexi Soft) 180
7.3.7 Saving and loading a configuration 181
7.3.8 Importing and exporting a configuration 181
7.4 Monitoring the operational data online 182

8 Technical specifications 183
8.1 Technical specifications gateways 183
8.1.1 EtherNet/IP, Modbus TCP, PROFINET IO 183
8.1.2 EtherCAT 183
8.1.3 PROFIBUS DP 184
8.1.4 CANopen 184
8.1.5 DeviceNet 185
8.2 Technical specifications, supply circuit 185
8.3 General technical specifications 186
8.4 Dimensional drawings 187
8.4.1 Dimensional drawing FX0-GENT, FX0-GMOD, FX0-GPNT and FX0-GETC 187
8.4.2 Dimensional drawing FX0-GPRO, FX0-GCAN and FX0-GDEV 188
8.5 Ordering information Flexi Soft gateways 188

9 Annex 189
9.1 EC declaration of conformity 189
9.2 List of tables 191
9.3 List of illustrations 195

1 About this document

Please read this chapter carefully before working with these operating instructions and the Flexi Soft gateways.

1.1 Function of this document

These operating instructions only apply in conjunction with the other Flexi Soft operating instructions (see section 1.2 “The Flexi Soft operating instructions” below). They provide the technical personnel at the machine manufacturer or machine operating organisation information on safe mounting, adjustment, electrical installation, commissioning as well as on operation and maintenance of the Flexi Soft gateways.

These operating instructions do not provide information on the operation of the machine in which a Flexi Soft modular safety controller and a Flexi Soft gateway is integrated. Information on this is to be found in the appropriate operating instructions for the machine.

1.2 The Flexi Soft operating instructions

For the Flexi Soft system there are three operating instructions with clearly distinguished fields of application as well as mounting instructions for each module.

- The mounting instructions (SICK part no. 8012482, 8012475, 8013272, 8012484 and 8013274) are enclosed with each Flexi Soft module. They inform on the basic technical specifications of the modules and contain simple mounting instructions. Use the mounting instructions when mounting Flexi Soft safety controllers.
- The Flexi Soft hardware operating instructions (SICK part no. 8012999) describe all Flexi Soft modules and their functions in detail. Use the Hardware operating instructions in particular to configure Flexi Soft safety controllers.
- The Flexi Soft gateways operating instructions (this document) describe all Flexi Soft gateways and their functions in detail.
- The Flexi Soft software operating instructions (SICK part no. 8012998) describe the software-supported configuration and parameterization of the Flexi Soft safety controllers. In addition, the software operating instructions contain the description of the diagnostics functions that are important for operation and detailed information for the identification and elimination of errors. Use the software operating instructions in particular for the configuration, commissioning and operation of Flexi Soft safety controllers.

1.3 Target group

These operating instructions are addressed to planning engineers, machine designers and the operators of systems in which a Flexi Soft modular safety controller is integrated and who want to exchange data with a fieldbus (a controller) via a gateway.

They are also addressed to people who are placing a Flexi Soft gateway in operation for the first time or maintaining it.

1.4 Information depth

These operating instructions contain information on the Flexi Soft gateways on the following subjects:

- mounting
- implementation into a network
- configuration via Flexi Soft Designer software
- data transfer to and from the network
- status information, planning and related mapping
- part numbers

WARNING

Warning!

Pay attention to the safety notes and safety measures on the Flexi Soft gateways!

Note We also refer you to our homepage on the Internet at

www.sick.com

There you will find the following files for download:

- FX0-GENT EDS file for EtherNet/IP
- FX0-GPNT GSDML file for PROFINET IO
- FX0-GPRO GSD file for PROFIBUS DP
- FX0-GCAN EDS file for CANopen
- FX0-GDEV EDS file for DeviceNet
- FX0-GETC ESI file for EtherCAT

1.5 Scope

These operating instructions are valid for all Flexi Soft gateway modules.

This document is part of SICK part number 8012662 (“Flexi Soft gateways” operating instructions in all available languages).

These operating instructions are the original operating instructions.

1.6 Abbreviations used

CSV Comma separated values

EFI Enhanced Function Interface

h Hexadecimal notation (e.g. 72h = 114)

SINT Short integer = 1 byte

UDINT Unsigned double integer = 4 bytes = 2 Words

UINT Unsigned integer = 2 bytes = 1 Word

USINT Unsigned short integer = 1 byte

1.7 Symbols used

Notes Notes provide special information on the device.

●, ☼, ○ LED symbols describe the state of a diagnostics LED. Examples:

- ● The LED is illuminated constantly.
- ☼ The LED is flashing.
- ○ The LED is off.

➢ **Action** Instructions for taking action are shown by an arrow. Read carefully and follow the instructions for action.

WARNING

Warning!

A warning notice indicates an actual or potential risk or health hazard. They are designed to help you to prevent accidents.

Read carefully and follow the warning notices!

1.8 Trademarks

Windows XP, Windows Vista, Windows 7 and Internet Explorer are registered trademarks of Microsoft Corporation in the USA and other countries.

SIEMENS SIMATIC Manager is a registered trademark of SIEMENS AG.

DeviceNet and DeviceNet Safety are registered trademarks of the Open DeviceNet Vendor Association, Inc. (ODVA).

TwinCAT is a registered trademark of Beckhoff Automation GmbH.

EtherCAT. "EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany."

Other product names and company names referenced in this manual are trademarks or registered trademarks of their respective companies.

2 On safety

This chapter deals with your own safety and the safety of the equipment operators.

➢ Please read this chapter carefully before working with a Flexi Soft gateway.

2.1 Qualified safety personnel

The Flexi Soft gateway must only be installed, commissioned and serviced by qualified safety personnel.

Qualified safety personnel are defined as persons who ...

- have undergone the appropriate technical training

and

- have been instructed by the responsible machine operator in the operation of the machine and the current valid safety guidelines

and

- have access to the operating instructions of the Flexi Soft gateway and Flexi Soft modular safety controller and have read and familiarised themselves with them.

2.2 Correct use

The Flexi Soft gateways can only be operated with a Flexi Soft system. The firmware version of the connected FX3-CPUx must be at least V1.11.0, the version of the Flexi Soft Designer configuration software must be at least 1.3.0.

The Flexi Soft gateways do not have a dedicated voltage supply.

WARNING

The Flexi Soft gateways are not suitable for operation on a safety fieldbus!

These gateways only generate non-safety-related fieldbus data (status bytes) for control and diagnostics purposes.

Do not use data from a Flexi Soft gateway for safety related applications!

With the Flexi Soft gateways it is possible to integrate non-safe data into the logic editor such that the safety function of the Flexi Soft system is compromised. Never implement the gateway into a Flexi Soft system without having this danger checked by a safety specialist.

These modules may only be used by qualified safety personnel and only on the machine where they have been installed and initialised by qualified safety personnel in accordance with the operating instructions.

WARNING

Pay attention to the safety notes and safety measures on the Flexi Soft gateways!

If the device is used for any other purposes or modified in any way — also during mounting and installation — any warranty claim against SICK AG shall become void.

Notes

- During the mounting, installation and usage of the Flexi Soft gateway, observe the standards and directives applicable in your country.

- The national/international rules and regulations apply to the installation, commissioning, use and periodic technical inspection of the Flexi Soft modular safety controller, in particular:
  - EMC directive 2004/108/EC,
  - Provision and Use of Work Equipment Directive 2009/104/EC,
  - the work safety regulations/safety rules.
- The operating instructions must be made available to the operator of the machine where a Flexi Soft system is used. The machine operator is to be instructed in the use of the device by qualified safety personnel and must be instructed to read the operating instructions.

WARNING

Use the Flexi Soft system in industrial environments only!

The Flexi Soft system complies, as per the “radiated emissions” generic standard, with the requirements of class A (industrial applications). The Flexi Soft system is therefore only suitable for use in an industrial environment.

UL applications

For UL applications the device must be used with a Class 2 power supply or Class 2 transformer in accordance with UL 1310 or UL 1585.

2.3 Environmental protection

The Flexi Soft gateways are designed for minimum impact on the environment, they consume only a minimum of energy and resources.

➢ At work, always act in an environmentally responsible manner.

2.3.1 Disposal

Unusable or irreparable devices should always be disposed as per the applicable national regulations on waste disposal (e.g. European waste code 16 02 14).

Note We would be pleased to be of assistance to you on the disposal of these devices. Contact us.

2.3.2 Separation of materials

WARNING

Only appropriately trained personnel are allowed to separate materials!

Caution is required when dismantling devices. There is a risk of injuries.

Before you send the devices for appropriate recycling, it is necessary to separate the different materials of the Flexi Soft gateways.

➢ Separate the housing from the rest of the parts (in particular the circuit board).

➢ Send the separated parts for recycling as appropriate (see Tab. 1).

Tab. 1: Overview on disposal by components

Components	Disposal
Product Housing, circuit boards, cables, connectors and electrical connecting pieces	Electronic recycling
Packaging Cardboard, paper	Paper/cardboard recycling

3 Product description Flexi Soft gateways

The Flexi Soft gateways allow the Flexi Soft System to send and receive non-safety related data to and from the external fieldbus system for control and diagnostics purposes.

Note In this manual, the data exchanged between the Flexi Soft system and the respective network will be considered always from the network master (PLC) point of view. Therefore data sent from the Flexi Soft system into the network will be referred to as *input data* while data received from the network will be referred to as *output data*.

WARNING

Do not operate a Flexi Soft gateway on a safety fieldbus!

The Flexi Soft gateway modules are not suitable for operation on a safety fieldbus. They do not support any safety mechanism, which would be mandatory to communicate within a safety network.

Configuration of the Flexi Soft gateways is performed using the Flexi Soft Designer configuration software on a PC or notebook connected to the FX3-CPUx over RS-232 interface or connected to the Ethernet gateways over Ethernet TCP/IP.

The safety relevant logic of the Flexi Soft system operates independently from the gateway. If however the Flexi Soft system has been configured to integrate non-safe information from the fieldbus into the logic editor, a decoupling of the gateway can result in availablity problems.

A Flexi Soft gateway can only be operated on a Flexi Soft system. It does not have a dedicated voltage supply. It is possible to use two Flexi Soft gateways per system.

The gateways are fitted in a 22.5 mm wide housing for 35 mm DIN mounting rails in accordance with EN 60 715.

Ordering information can be found in section 8.5 “Ordering information Flexi Soft gateways” on page 188.

3.1 Device variants

Six Flexi Soft gateways are available for the different network types. Suitable for Ethernet networks are the EtherNet/IP gateway FX0-GENT, the Modbus TCP gateway FX0-GMOD, the PROFINET IO gateway FX0-GPNT and the EtherCAT gateway FX0-GETC. PROFIBUS DP gateway FX0-GPRO, FX0-GCAN for CANopen and FX0-GDEV for DeviceNet are fieldbus gateways without Ethernet functionality.

Tab. 2: Device variants and features overview

Gateway	Network type	Ethernet TCP/IP socket interface	TCP/IP configuration interface
FX0-GENT	EtherNet/IP explicit messaging EtherNet/IP implicit messaging (only with firmware ≥ V2.00.0)	Client/server	Available at port 9000
FX0-GMOD	Modbus TCP master & slave receive method	Client/server	Available at port 9000
FX0-GPNT	PROFINET IO slave conformance class A	Client/server	Available at port 9000
FX0-GETC	EtherCAT slave	–	Available at port 9000 via EoE[1)]
FX0-GPRO	PROFIBUS DP slave	–	–
FX0-GCAN	CANopen slave	–	–
FX0-GDEV	DeviceNet-Slave	–	–

Note You will find the device's date of manufacture on the type label in the *S/N* field in the format yywwnnnn (yy = year, ww = calendar week, nnnn = continuos serial number in the calendar week).

3.2 Firmware versions

The Ethernet gateways FX0-GENT, FX0-GMOD and FX0-GPNT are available with different firmware versions. To add a gateway to a Flexi Soft system in the Flexi Soft Designer's hardware configuration window, you must select the appropriate step from the drop down list under the respective gateway.

Tab. 3: Firmware versions of the Ethernet gateways

Firmware version	Step
V1.xx.x	V1.xx
≥ V2.00.0	V2.xx

Notes

- You will find the firmware version on the type label of the device.
- If you use the Flexi Soft Designer command **Identify project** to identify a connected Flexi Soft system with a gateway, the software will recognize the correct firmware version automatically.

1) The TCP/IP configuration interface for the FX0-GETC is only available after configuring EoE (Ethernet over EtherCAT). See section 5.5.7 "Ethernet over EtherCAT (EoE)" on page 110 and section 5.5.8 "TCP/IP configuration interface" on page 110.

3.3 Data transmitted into the network (input data sets)

Available data

The Flexi Soft gateways can provide the following data:

- Operational data
  - **Logic results** from the Flexi Soft main module (FX3-CPUx) (see section 3.3.1 on page 17)
  - **Input values** (HIGH/LOW) for all Flexi Soft input extension modules in the system and EFI devices connected (see section 3.3.2 auf Seite 17)
  - **Output values** (HIGH/LOW) for all Flexi Soft input/output extension modules and EFI devices connected (see section 3.3.2 on page 17)
  - **Output data** from another network, i.e. data received by a second gateway in the Flexi Soft system (see section 3.3.4 on page 19)
  - **Gateway direct output values**: It is possible to write logic results directly from the logic editor into the gateway (see section 7.3.5 on page 178).
- Diagnostics data
  - **Checksums** (CRCs) (see section 3.3.5 on page 19)
  - **Error and status information** for all modules except the UE410-2RO and UE410-4RO (see section 3.3.6 on page 19)

Data sets

The physical Flexi Soft modules are not represented as typical hardware modules in the network. Instead, the data available from the Flexi Soft system has been organized into four *input data sets*.

- **Data set 1** (max. 50 bytes) contains the operational data. It can be compiled using the Flexi Soft Designer tool. Upon delivery there is a default selection for the content of data set 1 which can be freely modified. For details see Tab. 5 on page 16.

  For the FX0-GPNT and the FX0-GPRO, data set 1 has been subdivided in five *input data block*s, where data block 1-4 contain 12 bytes each and data block 5 contains two bytes. The FX0-GCAN holds four process data objects (PDOs) with 8 bytes each. For detailed information see the section on the related gateway.
- **Data set 2** (32 bytes) contains the system configuration CRCs. See Tab. 5 on page 16.
- **Data set 3** (60 bytes) contains the individual module status and diagnostics data with four (4) bytes per module. See section 3.3.6 “Error and status information of the modules” on page 19 for details.
- **Data set 4** (60 bytes) is currently filled with reserved values.

Tab. 4 gives an overview which data sets are available for each gateway.

Tab. 4: Availability of data set 1-4

	Data set 1	Data set 2	Data set 3	Data set 4
FX0-GENT	EtherNet/IP or TCP/IP	EtherNet/IP or TCP/IP	EtherNet/IP or TCP/IP	EtherNet/IP or TCP/IP
FX0-GMOD	Modbus TCP or TCP/IP	Modbus TCP or TCP/IP	Modbus TCP or TCP/IP	Modbus TCP or TCP/IP
FX0-GPNT	PROFINET IO or TCP/IP	PROFINET IO or TCP/IP	PROFINET IO or TCP/IP	PROFINET IO or TCP/IP
FX0-GPRO	PROFIBUS DP	–	–[2)]	–
FX0-GCAN	CANopen	CANopen (SDOs)	CANopen (SDOs)[3)]	–
FX0-GDEV	DeviceNet	DeviceNet	DeviceNet	DeviceNet
FX0-GETC	Input and output objects 2000h and 2001h	CRC's Object 2002h	Status and diagnostics Object 2003h	Reserved Object 2004h

2) With the FX0-GPRO, diagnostics data are available via PROFIBUS standard DP-V0 diagnostics. For more information on how to retrieve module status and diagnostics data via the PROFIBUS DP gateway please refer to chapter 6.1 "PROFIBUS DP gateway" on page 116.

3) With the FX0-GCAN, diagnostics data are available via CANopen SDO (service data objects) method. For more information on how to retrieve module status and diagnostics data via the CANopen gateway please refer to chapter 6.2 "CANopen gateway" on page 130.

Tab. 5: Overview input data sets 1-3 (default settings for EtherNet/IP, Modbus TCP and TCP/IP)

	Data set 1	Data set 2	Data set 3
Byte 0	Logic result 0	Overall CRC	Module status module 0. Module 0 is always the FX3-CPUx. For detailed information about the module status see section 3.3.6 “Error and status information of the modules” on page 19.
Byte 1	Logic result 1		
Byte 2	Logic result 2		
Byte 3	Logic result 3		
Byte 4	Input values module 1	System CRC (SCID)	Module status module 1
Byte 5	Input values module 2		
Byte 6	Input values module 3		
Byte 7	Input values module 4		
Byte 8	Input values module 5	Reserved	Module status module 2
Byte 9	Input values module 6		
Byte 10	Input values module 7		
Byte 11	Input values module 8		
Byte 12	Input values module 9	Verified configuration CRC	Module status module 3
Byte 13	Input values module 10		
Byte 14	Input values module 11		
Byte 15	Input values module 12		
Byte 16	Output values module 1	Reserved	Module status module 4
Byte 17	Output values module 2		
Byte 18	Output values module 3		
Byte 19	Output values module 4		
Byte 20	Output values module 5		Module status module 5
Byte 21	Output values module 6		
Byte 22	Output values module 7		
Byte 23	Output values module 8		
Byte 24	Output values module 9		Module status module 6
Byte 25	Output values module 10		
Byte 26	Output values module 11		
Byte 27	Output values module 12		
Byte 28	Gateway direct output values 0		Module status module 7
Byte 29	Gateway direct output values 1		
Byte 30	Gateway direct output values 2		
Byte 31	Gateway direct output values 3		
Byte ...	Not assigned		...
Byte 49	Not assigned		...
Byte ...			...
Byte 56			Module status module 14. Module 13 and 14 are always the gateways.
Byte 57			
Byte 58			
Byte 59			
Length	**50 bytes**	**32 bytes**	**60 bytes**

3.3.1 Logic results

Logic results generated by the logic editor of the Flexi Soft main module can be made available to the network. Up to 20 bytes are available where each bit represents one logic result from the logic editor. Data set 1 containing the logic results can be customized. For detailed information see the chapter on the related gateway and chapter 7 “Layout and content of the process image” on page 173.

3.3.2 Gateway direct output values

It is possible to write values directly from the logic editor into a gateway. There are four bytes reserved for this in the default settings for data set 1, however up to all 50 bytes of data set 1 can be defined as gateway direct output values. For more information please see section 7.3.5 “Gateway direct output values” on page 178.

Note In order to use gateway direct output values, a FX3-CPUx with firmware V2.00.0 or higher is required.

3.3.3 Module status and EFI status as well as input and output values

The Flexi Soft gateways can transmit the status and all input and output states of all Flexi Soft modules and EFI devices connected to the Flexi Soft system into the network. Data set 1 containing the input and output values and the EFI information can be customized. For detailed information see the chapter on the related gateway and chapter 7 “Layout and content of the process image” on page 173.

Module status

The Flexi Soft gateways can transmit the status of the modules connected into the network. A total of 6 bytes are available for this purpose.

Tab. 6: Module status

Module status	Size	Meaning	Allocation
Status input data	2 Byte	One sum bit per module for the status of the module’s inputs 0 = Error 1 = No error	Bit 0 = FX3-CPUx Bit 1 = 1. extension module Bit 2 = 2. extension module ... Bit 13 = 1. gateway Bit 14 = 2. gateway Bit 15 = reserved
Status output data	2 Byte	One sum bit per module for the status of the module’s outputs 0 = Error 1 = No error	
Position status	2 Byte	One sum bit per module for the status of the inputs and outputs (AND operator on input data status and output data status) 0 = Error 1 = No error	

You will find information on the significance of the status bits in the operating instructions for the software Flexi Soft Designer (SICK part no. 8012998) in the chapter “Module input and output status bits in the logic editor”.

Note The input and output status for the XTIO and XTDI modules is available only with firmware version V2.00.0 and higher.

Input values and output values of the modules

- Input values for I/O modules

  1 byte is available for data set 1 for each module.

  The input values indicate the status of the pre-evaluation on the I/O module. This status corresponds to the status of the element in the main module's logic. The level on the related terminal cannot be reliably detected by this means, as the data can be set low by the cross circuit detection or the dual-channel sampling independent of the level on the input terminal (e.g. I1-I8).

  If dual-channel input elements are configured on an I/O module, then the less significant bit represents the status of the pre-evaluation on the related 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 more significant bit (bit 1, 3, 5 and 7) is used as follows:

Tab. 7: Usage of the more significant bits on dual-channel evaluation on I/O modules FX3-XTIO

Firmware version FX3-XTIO	Dual-channel equivalent switches	Dual-channel complementary switches
V1.xx	Same status as the less significant bit	Inverted status of the less significant bit
V2.00 and higher	Status of the pre-evaluation 0 = Error 1 = No error	

- I/O modules output values

  For each module with outputs, 1 byte is available for data set 1.

  The output values indicate the status of the control information from the main module's logic for the related element on the I/O module. The level on the related terminals cannot be reliably detected from this status, as the output may be disabled by the cross circuit detection or the overload detection.

  If dual-channel output elements are configured on an I/O module, then only the less significant bit is used for 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 more significant bit (bit 1, 3, 5 and 7) is not used in this case (low).

- Input values for MOCx modules

  2 bytes are available for data set 1 for each MOCx module. The input values indicate the status of the signals from the main module's logic to the MOCx logic. Bit 16 and bit 17 of the usable bits from the main module to the MOCx logic are not available here.

- MOCx modules output values

  2 bytes are available for data set 1 for each MOCx module. The output values indicate the status of the signals from the MOCx logic to the main module's logic.

EFI system information

The FX3-CPU1 and FX3-CPU2 main modules have 2 EFI interfaces. An EFI interface is a safe communication interface between SICK devices. It allows to ...

- read out information from the safety devices (e.g. C4000, S3000).
- transfer commands to the safety devices.

The Flexi Soft gateways allow these EFI devices connected to the FX3-CPU1 or FX3-CPU2 to transmit their data into the network.

Note It is only possible to select the EFI data in byte arrays. 4 byte arrays for each connected EFI device are available. Some of the data content is reserved and can not be used at the PLC.

Further information about the properties, functions and benefits of the EFI interfaces can be found in the Flexi Soft Hardware operating instructions (SICK part no. 8012999).

The general EFI function description is available in the operating instructions “EFI - Enhanced Function Interface” (SICK part no. 8012622).

3.3.4 Routing of data from a second network

If your Flexi Soft system contains two gateways, it is possible to rout information received by the first gateway from one network (e.g. from a Modbus PLC) into a second network via the second gateway (e.g. to a PROFIBUS master) and vice versa.

3.3.5 Configuration checksums (CRCs)

Data set 2 contains the following configuration CRCs for the Flexi Soft system:

- overall CRC (same as system CRC). The overall CRC is the checksum displayed in the Flexi Soft Designer report.
- system CRC (SCID)
- verified configuration CRC. If this value is identical to the System CRC, the configuration is verified. Otherwise the configuration is not verified.

Each checksum is four bytes long. Data set 2 can not be customized.

3.3.6 Error and status information of the modules

Data set 3 contains the module status information transferred to the network.

Four bytes are transferred for each module (e.g. FX3-XTIO). Data set 3 can not be customized.

Notes

- Reserved (for future use) = static 1 (no status change)
- If no module is present, all values including the reserved values are set to logical 1.
- The four status bytes for each module are being transferred in Big Endian format as 32 bit word, i.e. the most significant byte (MSB = byte 3) is transferred first and the least significant byte (LSB = byte 0) is transferred last.

Module status bits of the FX3-CPUx main modules

The module status bits have the following significance, if not otherwise stated:

0 = Error

1 = No error

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	EFI2	EFI1	Power supply	Configuration Flexi Soft system	Reserved	Summary of bits 0.5 to 0.7 (power supply and EFI)	Internal tests	Module operating state 1 = Run 0 = Other
Byte 1	Reserved						Flexi Link stations suspended 1 = None 0 = One or more	Flexi Link stations in the system 1 = All found 0 = One or more are missing
Byte 2	Reserved							
Byte 3	Reserved							

Tab. 8: Module status bits of the FX3-CPUx main modules

Module status bits of the I/O modules FX3-XTIO and FX3-XTDI[4)]

The module status bits have the following significance, if not otherwise stated:

0 = Error

1 = No error

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Reserved	Output *Fast-Shut-Off*	Outputs power supply	Configuration of this module is valid.	Reserved	Summary of bits 0.5 to 0.7 (external error)	Internal tests	Module operating state 1 = Run 0 = Other
Byte 1	Reserved				Input 7-8 dual channel input evaluation	Input 5-6 dual channel input evaluation	Input 3-4 dual channel input evaluation	Input 1-2 dual channel input evaluation
Byte 2	External test signal Input 8	External test signal Input 7	External test signal Input 6	External test signal Input 5	External test signal Input 4	External test signal Input 3	External test signal Input 2	External test signal Input 1
Byte 3	Short-circuit monitoring output 4 Short-circuit to low	Short-circuit monitoring output 4 Short-circuit to high	Short-circuit monitoring output 3 Short-circuit to low	Short-circuit monitoring output 3 Short-circuit to high	Short-circuit monitoring output 2 Short-circuit to low	Short-circuit monitoring output 2 Short-circuit to high	Short-circuit monitoring output 1 Short-circuit to low	Short-circuit monitoring output 1 Short-circuit to high

Tab. 9: Module status bits of the I/O modules FX3-XTIO and FX3-XTDI

Module status bits of the I/O modules FX3-XTDS

The module status bits have the following significance, if not otherwise stated:

0 = Error

1 = No error

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Output load (overcurrent) monitoring	Reserved	Outputs power supply	Configuration of this module is valid.	Reserved	Summary of bits 0.5 to 0.7 (external error)	Internal tests	Module operating state 1 = Run 0 = Other
Byte 1	Reserved				Input 7-8 dual channel input evaluation	Input 5-6 dual channel input evaluation	Input 3-4 dual channel input evaluation	Input 1-2 dual channel input evaluation
Byte 2	External test signal Input 8	External test signal Input 7	External test signal Input 6	External test signal Input 5	External test signal Input 4	External test signal Input 3	External test signal Input 2	External test signal Input 1
Byte 3	Reserved							

Tab. 10: Module status bits of the I/O modules FX3-XTDS

Module status bits of the I/O modules FX0-STIO

The module status bits have the following significance, if not otherwise stated:

0 = Error

1 = No error

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Output load (overcurrent) monitoring	Reserved	Outputs power supply	Configuration of this module is valid.	Reserved	Summary of bits 0.5 to 0.7 (external error)	Internal tests	Module operating state 1 = Run 0 = Other
Byte 1	Reserved							
Byte 2	Reserved							
Byte 3	Reserved							

Tab. 11: Module status bits of the I/O modules FX0-STIO

[4)] The module status bits for the FX3-XTIO and FX3-XTDI are fully supported only with firmware version 1.2.x and higher.

Module status bits of the motion control modules FX3-MOCx

The module status bits have the following significance, if not otherwise stated:

0 = Error

1 = No error

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Reserved	Encoder 2	Encoder 1	Configuration of this module is valid.	Reserved	Summary of bits 0.5 to 0.7 (external error)	Internal tests	Module operating state 1 = Run 0 = Other
Byte 1	User-defined status bit 4[5)]	User-defined status bit 3[5)]	User-defined status bit 2[5)]	User-defined status bit 1[5)]	Reserved			
Byte 2	Reserved							
Byte 3	Reserved							

Tab. 12: Module status bits of the motion control modules FX3-MOCx

Module status bits of the gateways

The module status bits have the following significance, if not otherwise stated:

0 = Error

1 = No error

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Reserved	Communication to the network	Communication from the network	Configuration of this module is valid.	Reserved	Summary of bits 0.5 to 0.7 (external error)	Internal tests	Module operating state 1 = Run 0 = Other
Byte 1	Reserved							
Byte 2	Reserved							
Byte 3	Reserved							

Tab. 13: Module status bits of the gateways

Example

Module 2 (Flexi Soft XTIO) has a stuck-at-high error (24 V) on output 3. The following module status will be transferred into the network (only the first 12 of 60 bytes are shown):

Tab. 14: Example module status in data set 3

Byte address	**00**	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**...**
Byte	MSB			LSB	MSB			LSB	MSB			LSB	...
	3	2	1	0	3	2	1	0	3	2	1	0	...
Value	FF	FF	FF	FF	FF	FF	FF	FF	**EF**	FF	FF	**FB**	...
Meaning	Status module 0 (CPU)				Status module 1 (XTIO)				**Status module 2 (XTIO)**				...

The first relevant byte for the error on module 2 described above is the module status byte 0 for module 2. This is byte 11 with the hex value FB (11111011):

Tab. 15: Example module status byte 0 of module 2

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 Tab. 9).

[5)] The status of this bit can be defined to suit the specific application in the MOCx logic, e.g. to indicate inadmissible movements of an axis that have been detected by an MOCx function block.

The second relevant byte is the module status byte 3 for module 2. This is byte 08 with the hex value EF (11101111):

Tab. 16: Example module status byte 3 of module 2

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 output 3, short-circuit to high” (byte 3, bit 4 in Tab. 9).

You will find an example process image in section 5.1.3 “TCP/IP process image example” on page 46.

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

The data received from the network are organised in output data sets (max. 50 bytes). These data have been subdivided in five data blocks holding 10 bytes each for the FX0-GENT, FX0-GMOD, FX0-GPNT, FX0-GETC and the FX0-GDEV. For the FX0-GPRO output data blocks 1-4 hold 12 bytes each while output data block 5 holds 2 bytes. The FX0-GCAN has four process data objects (PDOs) with 8 bytes each.

Tab. 17: Output data blocks 1-5 for the different gateways

Gateway	Output data block 1 size	Output data block 2 size	Output data block 3 size	Output data block 4 size	Output data block 5 size
FX0-GENT	10 bytes	10 bytes	10 bytes	10 bytes	10 bytes
FX0-GMOD	10 bytes	10 bytes	10 bytes	10 bytes	10 bytes
FX0-GPNT	10 bytes	10 bytes	10 bytes	10 bytes	10 bytes
FX0-GETC	10 bytes	10 bytes	10 bytes	10 bytes	10 bytes
FX0-GPRO	12 bytes	12 bytes	12 bytes	12 bytes	2 bytes
FX0-GCAN	8 bytes	8 bytes	8 bytes	8 bytes	–
FX0-GDEV	10 bytes	10 bytes	10 bytes	10 bytes	10 bytes

The contents of the output data blocks can be used within the FX3-CPUx logic editor and can also be made available to another network via a second Flexi Soft gateway in the Flexi Soft system.

Notes

- In order to make the data from the network available in the logic editor or as input to another network, you must define a tag name for each bit that shall be used.
- Bits without a tag name will not be available in the logic editor nor for routing via a second gateway. For detailed information on how to define tag names for the data received please see the related section in the chapters on the different gateways.
- The status of the communication to and from the network can be monitored in the logic editor using the module input status bit for data from the network and the module output status bit for data to the network. When the gateway detects an invalid communication, the contents of the data sets will be set to zero (logical 0) and the corresponding module status bit will also be set to zero (logical 0).
- In case all communication is dropped, the data of the output data sets will be set to zero (logical 0) and the module input status bit will also be set to zero (logical 0).
- If a connection is closed while still others are available, the MS LED or the STATUS LED on the related gateway flashes Red/green for 10 s and an entry is made in the error history. In this case, the status bits will not be affected.

WARNING

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

The output data set can be written to the Ethernet gateways in parallel by all communication interfaces or TCP/IP sockets (e.g. Modbus TCP and Ethernet TCP/IP), if they use the same output data set number. In that case, the last message overrides data received earlier.

4 Mounting and basic configuration of the gateways

4.1 Mounting/Dismantling

This chapter describes the mounting of the Flexi Soft gateways.

WARNING

Make sure that the connection of the Flexi Soft gateway cannot lead to hazardous situations during installation!

Ensure that connecting a Flexi Soft gateway cannot lead to a hazardous situation when implementing the unit on to the Flexi Soft system and Ethernet network. Prevent unintended start-up of equipment during connection of a Flexi Soft gateway.

4.1.1 Steps for mounting the modules

WARNING

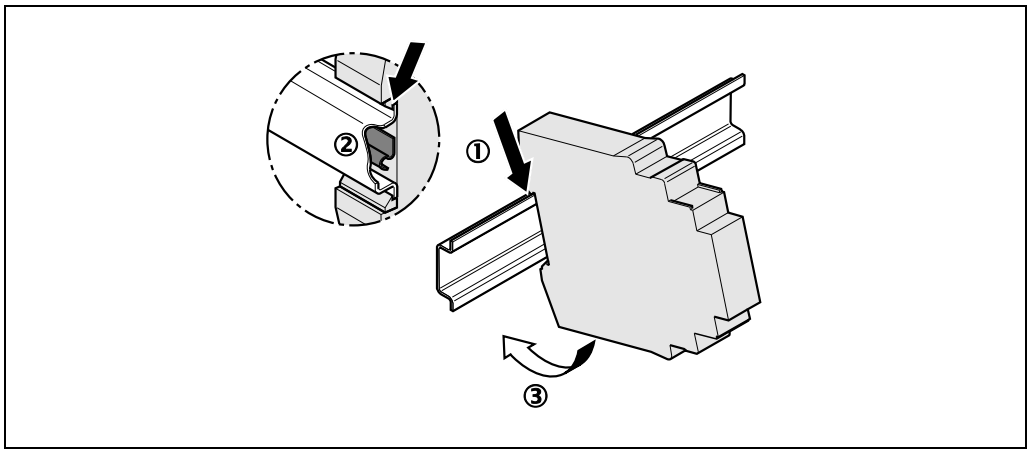
The Flexi Soft system is only suitable for mounting in a control cabinet with at least IP 54 enclosure rating.

While supply voltage is applied, modules must not be plugged to nor be removed from the Flexi Soft system.

To ensure full electromagnetic compatibility (EMC), the DIN mounting rail must be connected to functional earth (FE). Additionally connect all network cable shields directly at the control cabinet entrance to a common FE ground line.

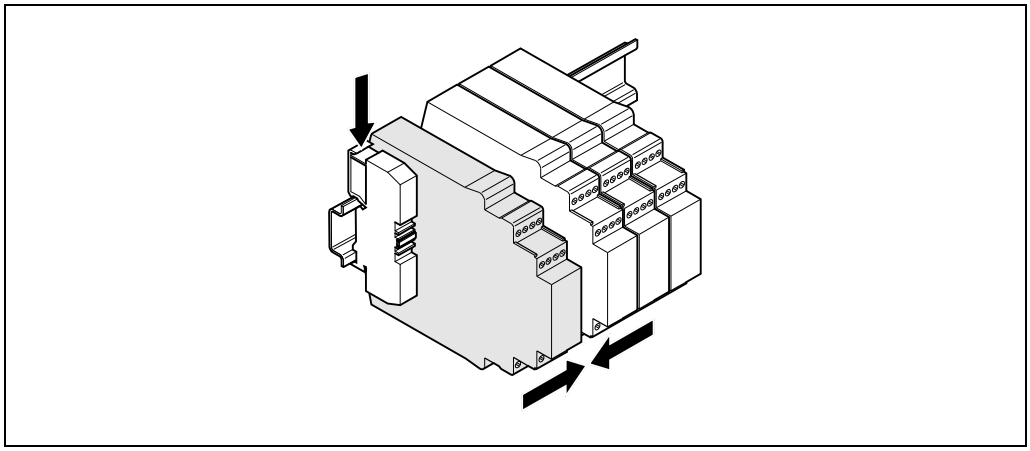
- In a Flexi Soft system the main module FX3-CPUx is positioned at the extreme left.
- The two optional gateways follow directly to the right of the main module.
- Connect further Flexi Soft extension modules (e.g. FX3-XTIO or FX3-XTDI) onto the right side of the gateways and any additional relay modules (UE410-2RO or UE410-4RO) to the extreme right of the entire Flexi Soft system.
- Ensure that suitable ESD protective measures are taken during mounting. Otherwise the devices may be damaged.
- The connection between the modules is effected by means of the plug connection integrated in the housing. Take into account that, when replacing a module, the Flexi Soft modules have to be pushed approx. 10 mm apart before the corresponding module can be removed from the DIN mounting rail.
- Take suitable measures to ensure that foreign matter does not penetrate the connector openings, in particular that of the system plug.
- Mount the modules in accordance with EN 50 274.
- The modules are located in a 22.5 mm wide modular system for 35 mm DIN mounting rails according to EN 60 715.

Fig. 1: Mounting the module onto the DIN mounting rail



- Make sure that the voltage supply of the Flexi Soft system is switched off.
- Hang the device onto the DIN mounting rail (①).
- Connect the gateways directly onto the right side of the FX3-CPUx module of the Flexi Soft system. Up to two gateways per system are possible.
- Ensure that the earthing spring contact (②) contacts the DIN mounting rail such that it can electrically conduct.
- Latch the module onto the DIN mounting rail by pressing it lightly in the direction of the arrow (③).

Fig. 2: Installing the end clips



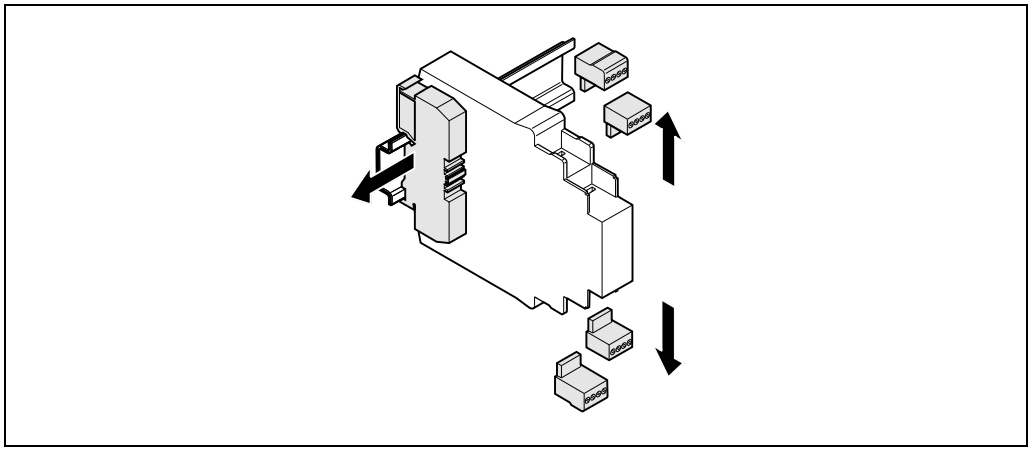
- If there are several modules, slide the modules together individually in the direction of the arrow until the side plug connection latches in.
- Install end clips on the left and right.

The following steps are necessary after mounting:

- Complete the electrical connections (see section 4.2 “Electrical installation” on page 27)
- Configuration (see section 4.3 “First configuration steps” on page 27 and the Flexi Soft Designer operating instructions, SICK part no. 8012998).
- Checking the installation (see the chapter on commissioning in the Flexi Soft hardware operating instructions, SICK part no. 8012999).

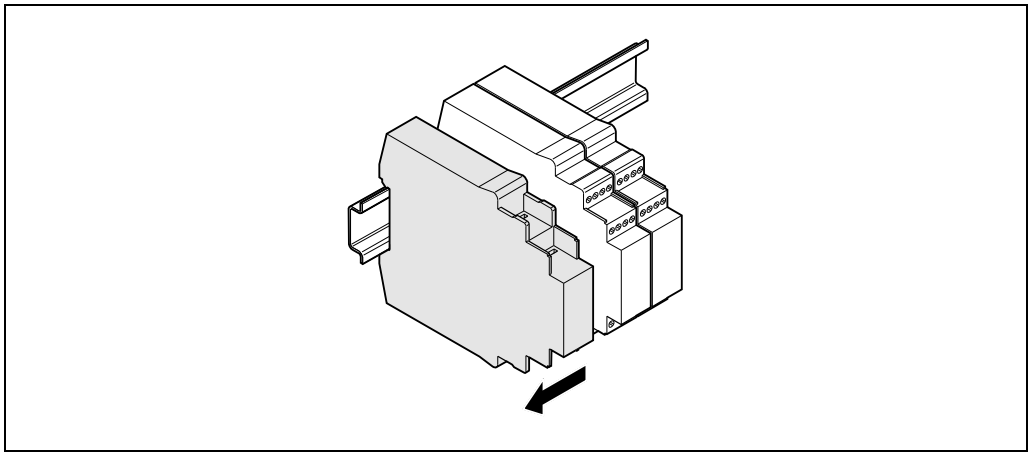
4.1.2 Steps for dismantling the modules

Fig. 3: Removing the removable terminals



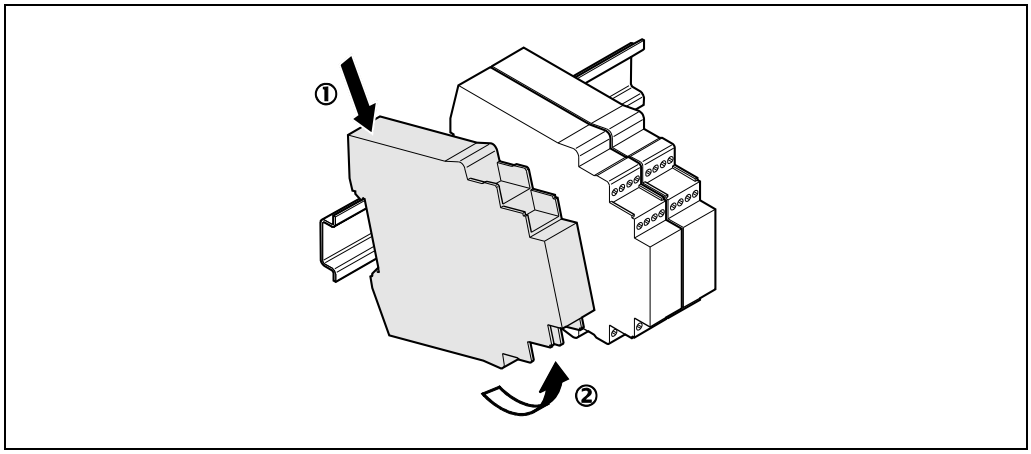
➢ Remove the removable terminals with the wiring and the end clips.

Fig. 4: Disconnecting the plug connections



➢ If there are several modules, slide the modules away from each other individually in the direction of the arrow until the side plug connection is separated.

Fig. 5: Removing modules from the DIN mounting rail



➢ Press the module downwards at the rear (①) and remove it from the DIN mounting rail in the direction of the arrow while keeping it pressed down (②).

4.2 Electrical installation

WARNING

Switch the entire machine/system off line!

The system could start up unexpectedly while you are connecting the devices.

Notes

- The Flexi Soft gateways fulfil the EMC requirements in accordance with the basic specification EN 61000-6-2 for industrial use.
- To ensure full electromagnetic compatibility (EMC), the DIN mounting rail must be connected to functional earth (FE).
- The control cabinet or assembly casing of the Flexi Soft system must comply at least with enclosure rating IP 54.
- Mounting in accordance with EN 50274.
- Electrical installation in accordance with EN 60204-1.
- The voltage supply of the devices must be capable of buffering brief mains voltage failures of 20 ms as specified in EN 60204-1.
- The voltage supply as well as all signals connected have to fulfil the regulations for extra-low voltages with safe separation (SELV, PELV) in accordance with EN 60664 and EN 50178 (equipment of electrical power installation with electronic devices).
- Ensure that all the modules of the Flexi Soft system, the connected protective devices as well as the voltage supplies are connected with the same GND plane. The GND of the RS-232 interface is connected internally to the GND of the supply of the main module (A2).
- Connect all fieldbus and Ethernet cable shields directly at the control cabinet entrance to the functional earth (FE).

4.3 First configuration steps

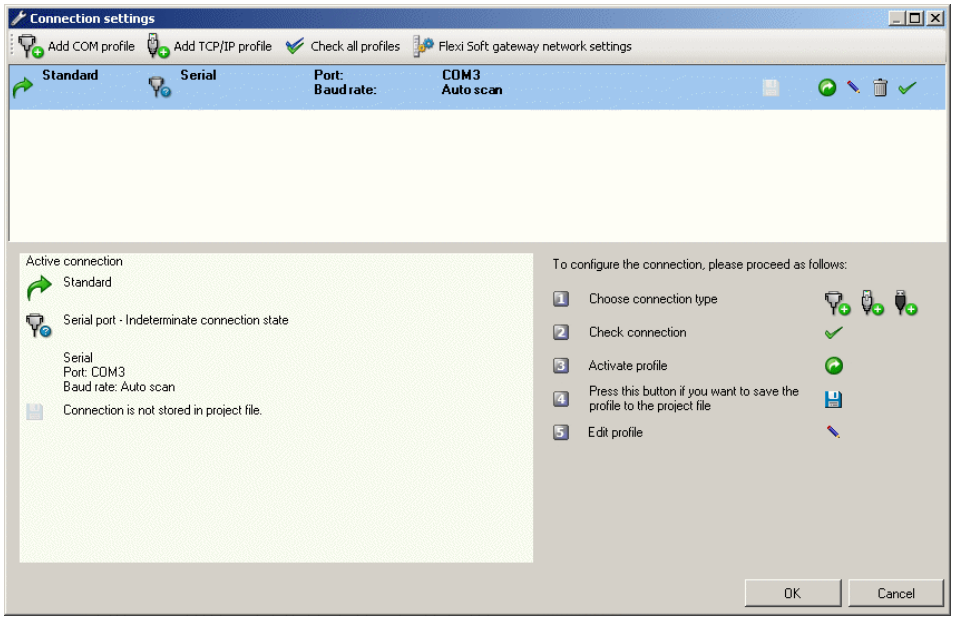
This chapter describes the basic steps you have to perform for the configuration of the gateway:

- establish a first connection between the gateway and a PC or notebook
- upload or transfer of a configuration
- verification of a configuration

4.3.1 Establishing a connection between gateway and PC

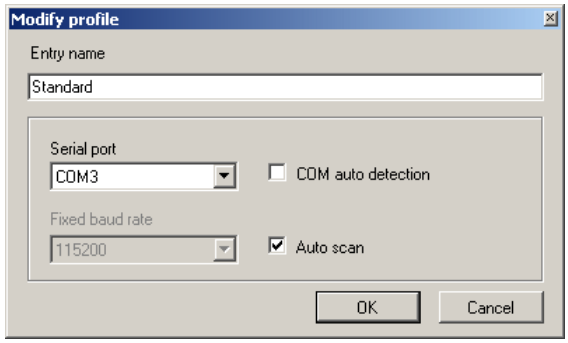
- Connect a PC or notebook to the RS-232 interface of the FX3-CPUx.
- Power up the Flexi Soft System.
- Open the Flexi Soft Designer configuration tool installed on the PC.
- Click on **Edit com. interface settings** to ensure the correct communication interface has been selected. The following dialog appears:

Fig. 6: Connection settings dialog



- To edit the settings, click on the pencil icon to the right. The following dialog appears:

Fig. 7: Modify profile dialog



- Modify the settings if required and click **OK**.
- Click **OK**. The dialog closes.
- Click on **Connect to physical device**. The Flexi Soft Designer will search for connected Flexi Soft devices and load the hardware configuration into the hardware configuration dialog. Once all modules have been identified correctly, the Flexi Soft Designer will ask whether the configuration shall be uploaded.
- Click **Yes** to upload the configuration.

As an example, the following hardware configuration may appear:

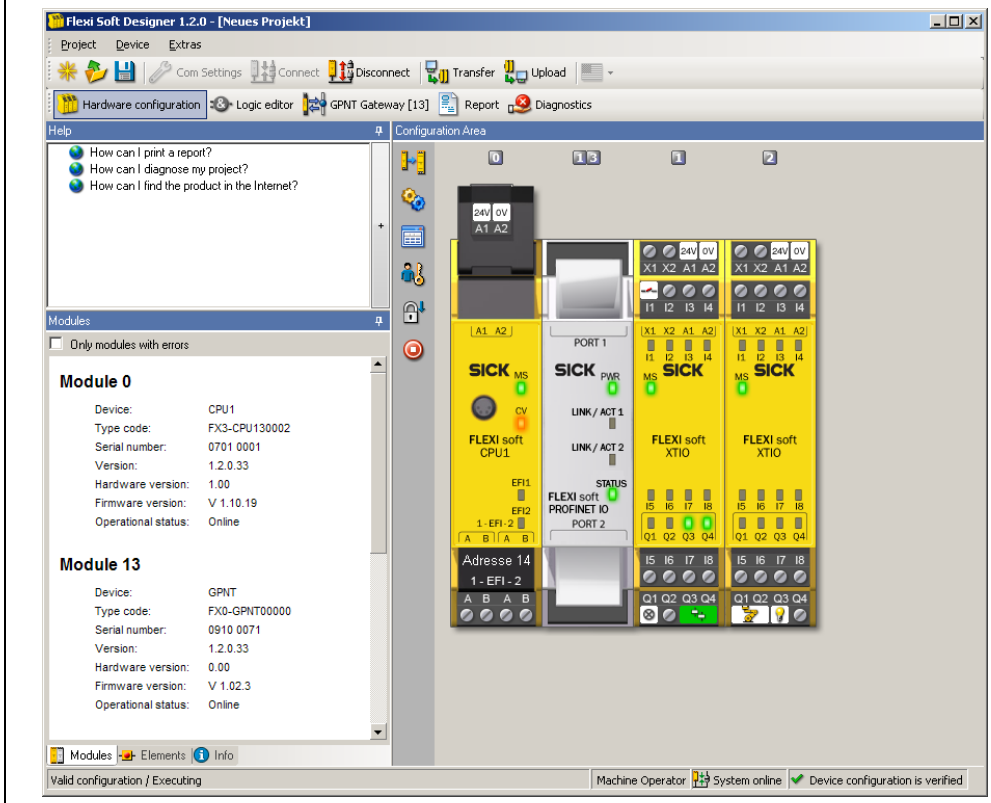


Fig. 8: Hardware configuration dialog

➢ Click **Disconnect** to go into the offline mode if you want to change the configuration of the Flexi Soft modules.

4.3.2 Configuration of the gateways

For the configuration of the gateways please refer to the sections on the related gateway:

- section 5.2 “EtherNet/IP gateway” on page 52
- section 5.3 “Modbus TCP gateway” on page 75
- section 5.4 “PROFINET IO gateway” on page 84
- section 5.5 “EtherCAT gateway” on page 101
- section 6.1 “PROFIBUS DP gateway” on page 116
- section 6.2 “CANopen gateway” on page 130

For the configuration of the TCP/IP interface of the Ethernet gateways, please refer to the following sections:

- section 5.1.1 “TCP/IP configuration interface” on page 32
- section 5.1.2 “Ethernet TCP/IP socket interface” on page 37

For the configuration of the operational data (data transfer from and to the network), please refer to chapter 7 “Layout and content of the process image” on page 173.

More information can be found in the Flexi Soft Designer operating instructions (SICK part no. 8012998).

4.3.3 Transfer of a configuration

Once you have finished the configuration, you have to transfer the configuration to your Flexi Soft system. In order to transfer a configuration, perform the following steps:

- Click **Connect** to go online. The Flexi Soft Designer connects to the Flexi Soft system.
- Click **Transfer** to transfer the configuration to the Flexi Soft system.

Note Depending on your current user level, you will be prompted to log on as authorized client to be able to transfer a configuration. For details please see the Flexi Soft Designer operating instructions (SICK part no. 8012998).

- Once the transfer has been completed, you will be asked whether you want to run the CPU module. Depending on your choice, click **Yes** or **No** to leave the dialog.

Note You can also start and stop the application in the **Hardware configuration** view using the **Run application** or **Stop application** buttons while the project is online.

More information can be found in the Flexi Soft Designer operating instructions (SICK part no. 8012998).

4.3.4 Verification of a configuration

After the configuration has been transferred successfully, the Flexi Soft system can be verified. To this purpose, the downloaded configuration data are read back out from the Flexi Soft system and compared with the project data. If they match, the data are displayed in a report. If the user confirms that they are correct, the system is considered to be verified.

- In the **Hardware configuration** view, click on the **Upload and verify configuration** button. A report of the current configuration will be generated.
- Click Yes below at the question **Mark device as verified?** if the displayed configuration is the expected configuration. The system is then considered to be verified.

Notes

- You have to be logged in as authorized user in order to mark the configuration as “verified”.
- If the verification is completed successfully, a “Read in and compare” report that provides the most important project information is created subsequently. You can print out or store this report.
- The status verified/not verified is indicated in the lower right-hand corner of the Flexi Soft Designer and by the CV LED at the Flexi Soft main module lighting up.
- Only if the device and the corresponding configuration have been marked as verified, the “Auto Start mode” is active in the configuration of the main module. If the configuration is not set to verified, the system stays in Idle mode (CV LED on the FX3-CPUx module flashing) after power-up and needs to be set to Run state using the Flexi Soft Designer.
- If differences between the project data and the read-back configuration data are detected, a corresponding message including information about possible actions is displayed. Verification of the configuration is not possible then. Observe the information in the error message for the further procedure. Terminate the dialog box by clicking **Close**.
- If you change a verified configuration, the status is reset to “not verified”.

  Exception: If you make only non safety-related changes such as modifying the gateway name, the gateway’s IP address or the port number for a TCP/IP socket connection, the configuration status remains “verified”.

More information can be found in the Flexi Soft Designer operating instructions (SICK part no. 8012998).

4.3.5 Upload of a configuration

When in online mode, you can upload a configuration from the connected Flexi Soft system:

➢ Click on **Upload**. The current configuration of the Flexi Soft system will be loaded into the Flexi Soft Designer and can be edited after going offline.

5 Ethernet gateways

This chapter describes the following Flexi Soft gateways:

- EtherNet/IP gateway (FX0-GENT)
- Modbus TCP gateway (FX0-GMOD)
- PROFINET IO gateway (FX0-GPNT)
- EtherCAT gateway (FX0-GETC)

5.1 Common features of the Ethernet gateways

5.1.1 TCP/IP configuration interface

The Flexi Soft Ethernet gateways offer a TCP/IP configuration interface which allows the configuration of the Flexi Soft system over Ethernet TCP/IP. This runs parallel to the Ethernet TCP/IP or other Ethernet protocols.

WARNING

Do not connect to the Flexi Soft system via the RS-232 and the Ethernet interface at the same time!

The Flexi Soft system can only communicate with one instance of the Flexi Soft Designer at one time. Connecting to the Flexi Soft system using multiple instances of the Designer, either on a single PC or multiple PCs, may result in inconsistencies of the configuration and the diagnostics as well as in operational errors. This applies to both RS-232 and Ethernet connections equally.

WARNING

Consider the signal delay for remote TCP/IP connections!

Remote TCP/IP connections to the gateway may be unstable if the signal delay is too great.

➢ Check the signal delay to the gateway using the ping command. Signal delays > 300 ms may cause the connection to break down.

Possible solutions:

➢ Make sure that the connection is fast enough or make another route if this is possible.

Or:

➢ Use a software tool like *Teamviewer* or *PC anywhere* to control a local computer on which the Flexi Soft Designer software is installed and connected with the Flexi Soft system.

Or:

➢ Contact SICK support.

In order to configure a gateway for TCP/IP configuration for the first time, perform the following steps:

Step 1: Assign an IP address

➢ Connect a PC or notebook to the RS-232 interface of the FX3-CPUx.

➢ Power up the Flexi Soft System.

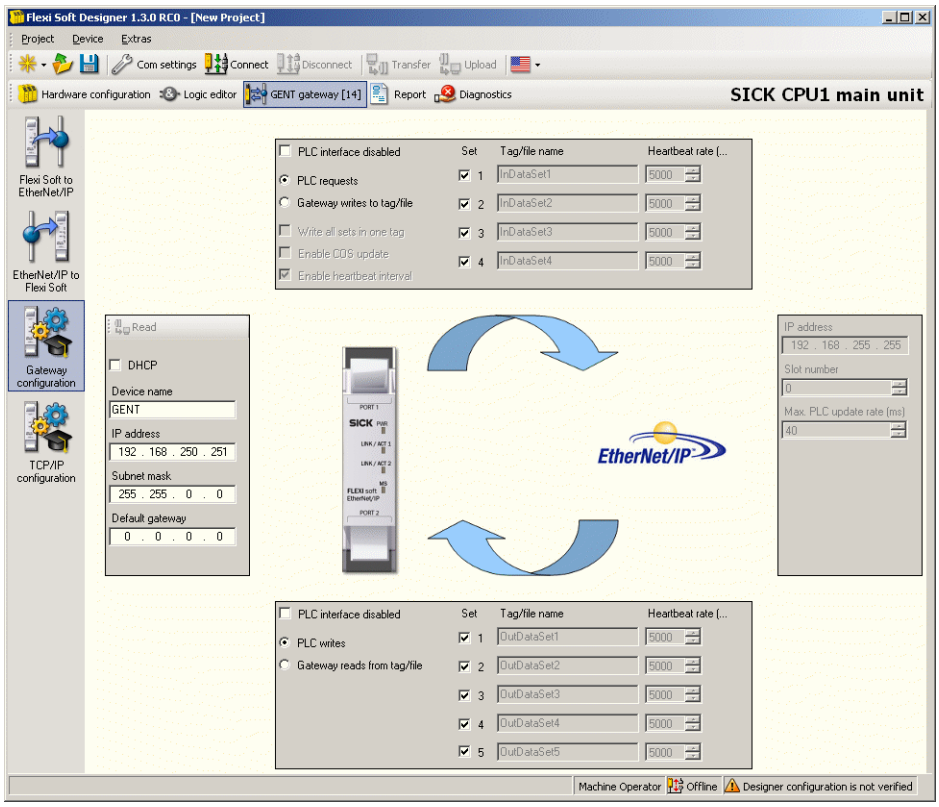
➢ Open the Flexi Soft Designer configuration tool installed on the PC and load the hardware configuration including the gateway.

➢ If your project is online, click on the **Disconnect** button to go offline.

➢ Click on the **Gateways** button above the main window and select the desired gateway or double click on the desired gateway in the hardware configuration view. The gateway configuration view opens.

➢ Click on the **Gateway configuration** button on the left hand menu. The following dialog appears:

Fig. 9: Ethernet gateway configuration dialog



On the left side of the dialog you will find the area for the gateway's IP configuration.

➢ If desired, enter a **Device name** for the Flexi Soft gateway.

➢ Enter a valid **IP address**, for the Flexi Soft gateway, and if required a valid **Subnet mask** and a valid IP address for a **Default gateway**.

Or:

➢ If your network uses a DHCP server, activate the **DHCP** checkbox.

➢ Click **Connect** to go online and transfer the configuration to the Flexi Soft system.

Notes:

- If your project is online, you can use the **Read IP address** button at the upper left corner of the gateway IP configuration area to retrieve the current IP settings of the gateway.
- The out-of-the-box default IP address of the gateway is 192.168.250.250. You can find the default IP address also on the type label of the gateway.

Step 2: Add a TCP/IP profile to your project

➢ Connect one of the two Ethernet ports of the gateway with your Ethernet network using a shielded Ethernet cable.

➢ Connect a PC (or notebook) to the same Ethernet network. Ensure the IP address settings of the PC match the network setup.

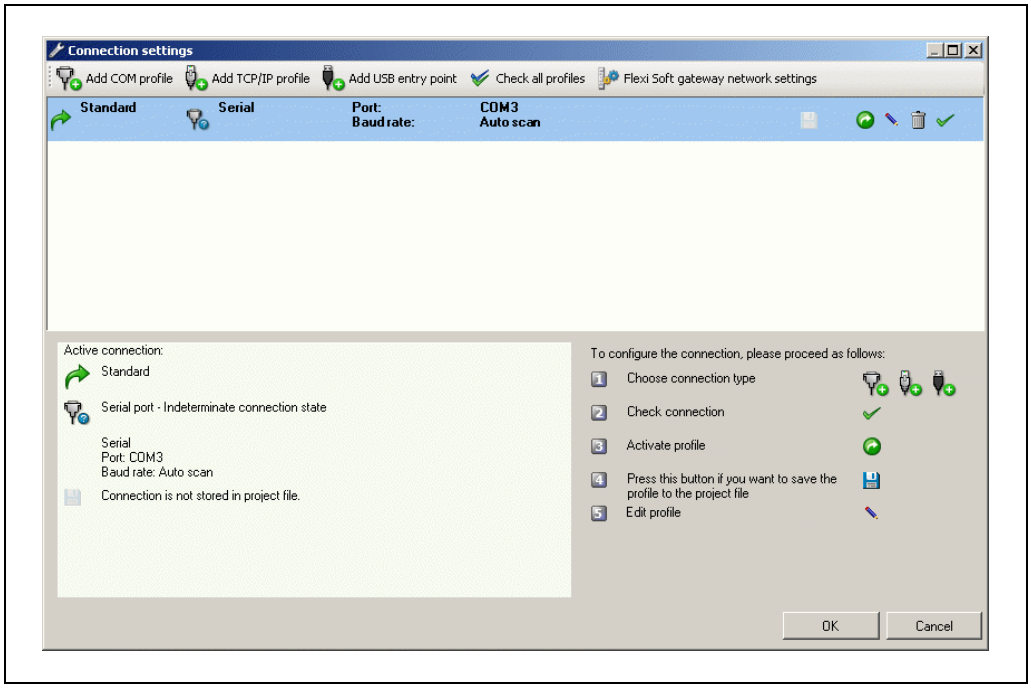
Note You can also connect your PC directly to one of the two Ethernet ports of the gateway. In this case, you can either adapt the IP address settings of your PC or the IP address settings of the gateway to match the other device's IP setup.

➢ Open the Flexi Soft Designer configuration tool installed on the PC and upload the hardware configuration including the gateway.

➢ If your project is online, click on the **Disconnect** button to go offline.

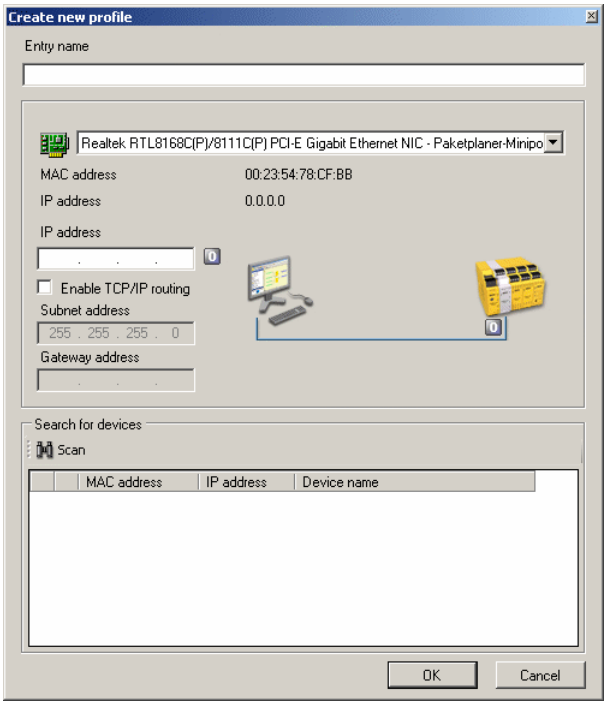
➢ Click on **Com settings**. The following dialog appears:

Fig. 10: Connection settings dialog



➢ Click on **Add TCP/IP profile**. The following dialog appears:

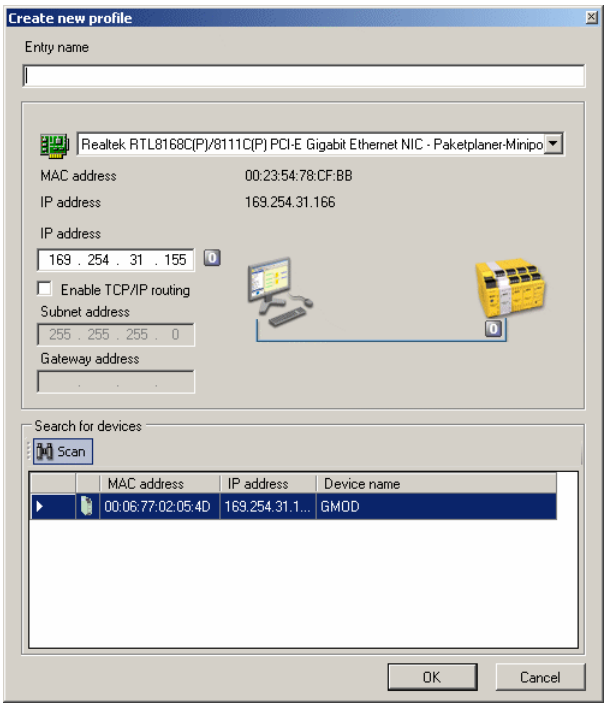
Fig. 11: Create new TCP/IP profile dialog



➢ Select your Ethernet network adaptor from the drop down list.

➢ Click on **Scan** to search for Flexi Soft gateways on your Ethernet network. Gateways located will be displayed as shown in the dialog below. Their IP address will be displayed as well as their MAC address and device name.

Fig. 12: Create new TCP/IP profile dialog after scan has been performed



Notes

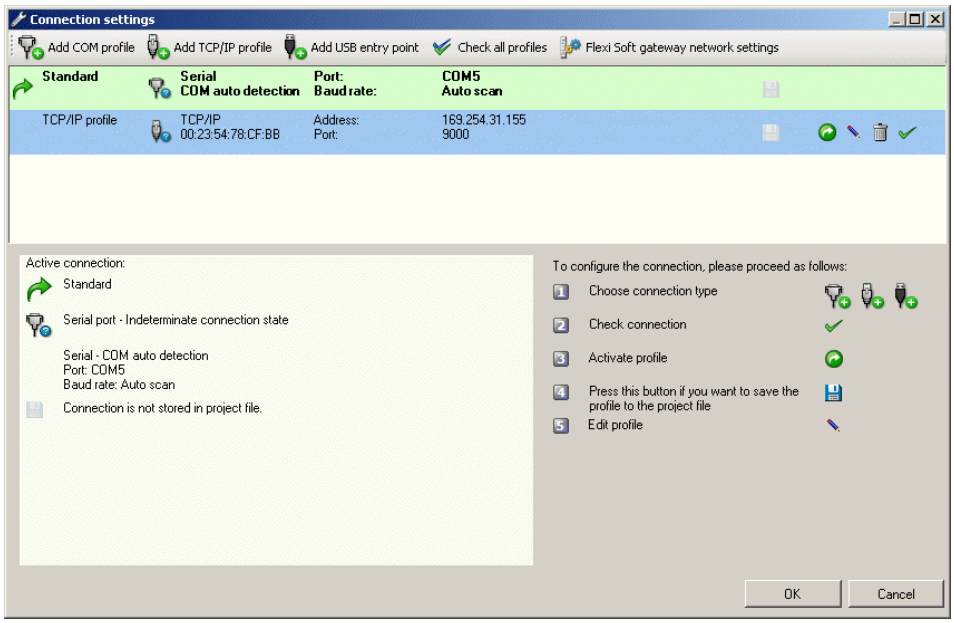
- Flexi Soft Designer with version ≥ 1.4.0 performs an UDP scan. This means that all Flexi Soft Ethernet gateways with firmware version ≥ V2.00.0 (FX0-GMOD, FX0-GPNT and FX0-GENT) in the network will be detected, even if they are in a different subnet.
- A FX0-GETC can be detected as well, if it has been configured for EoE before (see section 5.5.7 “Ethernet over EtherCAT (EoE)” on page 110 and section 5.5.8 “TCP/IP configuration interface” on page 110).
- Flexi Soft Designer version < 1.4.0 can detect only gateways with a matching subnet address.

➢ Select the gateway that you want to use for the new profile.

➢ Enter a name for the profile to the **Entry name** edit field.

➢ Click **OK**. The profile has now been created and is shown in the connection dialog:

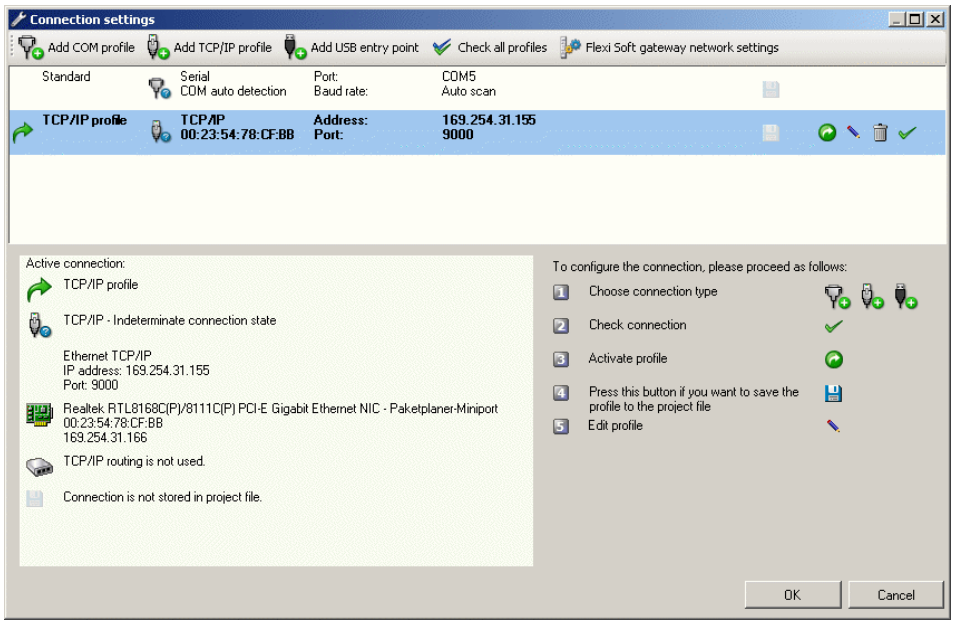
Fig. 13: Connection settings dialog with new TCP/IP profile



In order to use this profile, it needs to be activated.

➢ Click on the **Activate profile** icon (white arrow in green circle) on the far right. The profile will then be activated and marked as such:

Fig. 14: Connection settings dialog with new TCP/IP profile activated



➢ Click **OK**. All communication to the Flexi Soft system will now happen via TCP/IP. In order to use the profile via the serial interface again, you will have to re-activate it.

Note The port number for the TCP/IP configuration interface is pre-set to port 9000 and can not be changed.

Step 3: Connect via TCP/IP

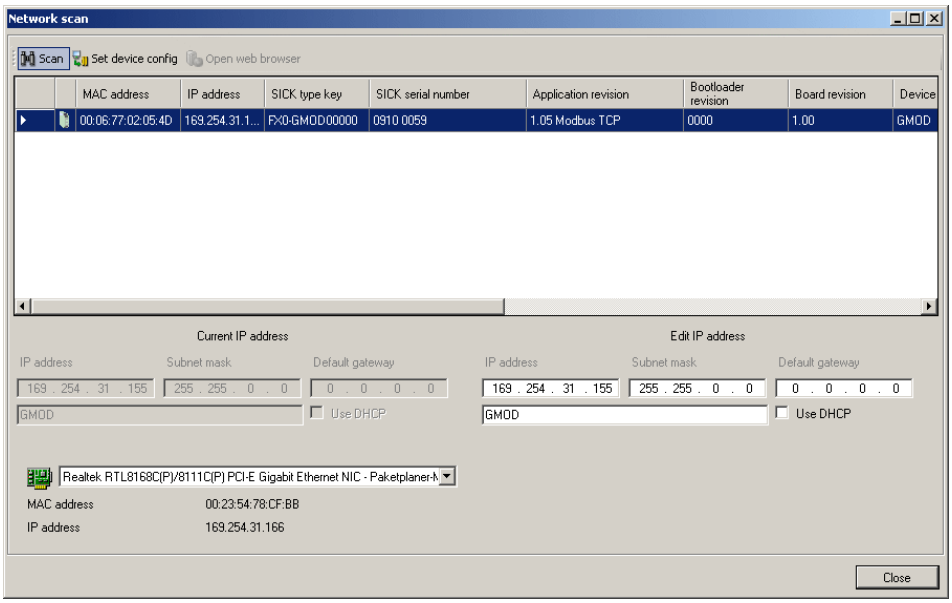
➢ Click on the **Connect** button to go online.

How to change the network settings of a Flexi Soft gateway:

➢ Click on the **Flexi Soft gateway network settings** button. The **Network scan** dialog is opened.

➢ Click on the **Scan** button. Your network is scanned for connected gateways and the devices found are displayed in the list.

Fig. 15: Found gateways in the Network scan dialog



Notes

- Flexi Soft Designer with version ≥ 1.4.0 performs an UDP scan. This means that all Flexi Soft Ethernet gateways with firmware version ≥ V2.00.0 (FX0-GMOD, FX0-GPNT and FX0-GENT) in the network will be detected, even if they are in a different subnet.
- A FX0-GETC can be detected as well, if it has been configured for EoE before (see section 5.5.7 “Ethernet over EtherCAT (EoE)” on page 110 and section 5.5.8 “TCP/IP configuration interface” on page 110).
- Flexi Soft Designer version < 1.4.0 can detect only gateways with a matching subnet address.

➢ Click on the gateway you want to edit.

➢ Enter the new settings in the **Edit IP address** area.

➢ Click on the **Set device config** button to transfer the new settings to the device.

Note If the Flexi Soft Designer identifies a Flexi Classic series gateway in the network, this will be displayed in the list as well. These gateways are equipped with an internal web server and can be addressed using the **Open web browser** button.

5.1.2 Ethernet TCP/IP socket interface

The FX0-GENT, FX0-GMOD and FX0-GPNT Ethernet gateways support a total number of four TCP/IP socket interfaces each. This allows multiple applications to communicate with the gateway at the same time over Ethernet TCP/IP. The number of possible connections depends on the gateway’s firmware version.

Tab. 18: Number of possible TCP/IP connections

Firmware version	Possible connections per socket	Possible connections overall
V1.xx.x	1 per socket	4
≥ V2.00.0	6 per socket	24

The gateway's proprietary network interface (e.g. Modbus TCP) runs in parallel and its configuration or usage does not interact with the TCP/IP socket configuration as it happens independently on separate Flexi Soft Designer pages.

WARNING

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

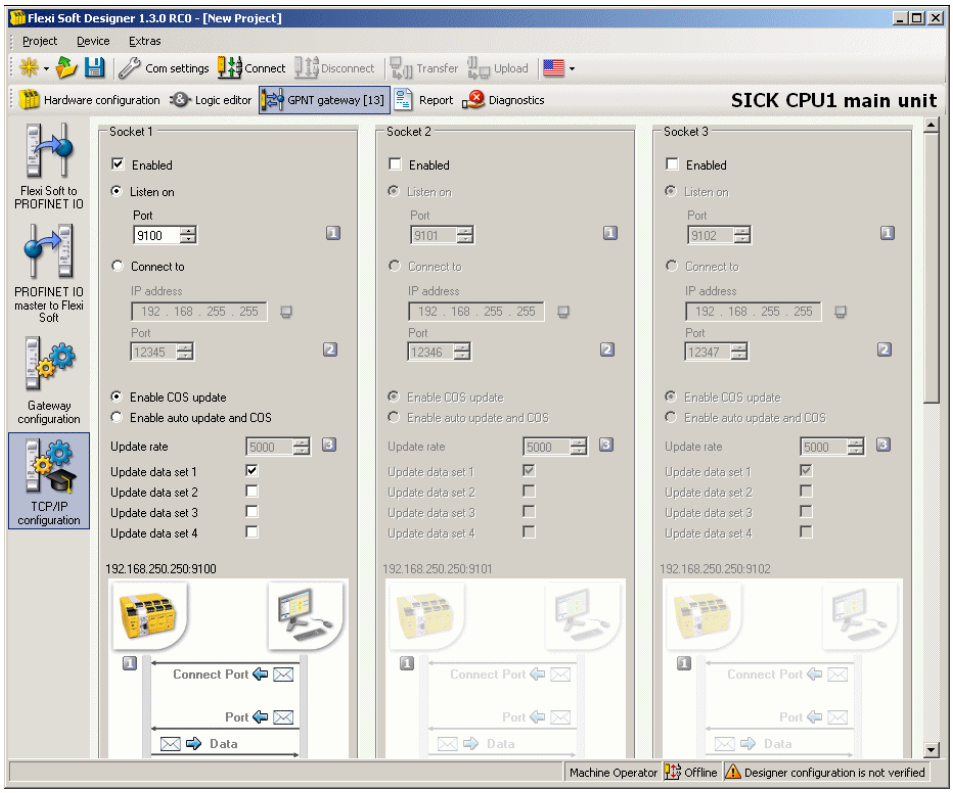
The output data set can be written to the Ethernet gateways in parallel by all communication interfaces or TCP/IP sockets (e.g. Modbus TCP and Ethernet TCP/IP), if they use the same output data set number. In that case, the last message overrides data received earlier.

The gateway processes the data of a Flexi Soft system and makes it available in different compilations, the *data sets*. These data sets are available over the TCP/IP interface. For a detailed description of the data sets please refer to section 3.3 "Data transmitted into the network (input data sets)" on page 14.

In order to configure the Ethernet TCP/IP socket interface, perform the following steps:

- Open the Flexi Soft Designer and load the hardware configuration including the gateway.
- Click on the **Gateways** button above the main window and select the desired gateway or double click on the desired gateway in the hardware configuration view to open the gateway configuration dialog.
- Click on **TCP/IP configuration** on the left hand menu. The following dialog appears:

Fig. 16: TCP/IP configuration dialog



Configuration of the TCP/IP interface — Who establishes the connection

If the Flexi Soft gateway shall connect to external applications, perform the following configuration steps:

- Activate the **Connect to** radio button.
- Set **IP address** to the IP address of the computer the application is running on.
- Enter the **Port** number for the application.

Note The configuration is considered faulty if either the connect socket port and/or the connect IP address is zero when in **Connect** mode.

If external applications shall connect to the Flexi Soft gateway, perform the following configuration steps:

- Activate the **Listen on** radio button.
- Enter the **Port** number for the application.

Notes
- Suggested port numbers are 9100 to 9103 (default values).
- Port 0 and port 9000 are reserved and can not be used (faulty configuration).
- Port numbers 0 to 1023 are managed by the Internet Assigned Numbers Authority (IANA) and should **not** be used to prevent collisions. See www.iana.org/assignments/port-numbers.

Finally, determine how the data are transferred. Follow the steps outlined in the following section.

Data transfer method — How the data are transferred

Whenever the TCP/IP socket connection has been established (either by an application on a PC or by the gateway itself), there are two possible methods how the data sets can be transferred:

- The application requests the data set(s) per command message (Application requests (Polling) mode),

  or

- the gateway auto-updates the data sets as per configuration (Gateway writes to Address/Port (Auto update) mode).

In Auto update mode there are two update modes how the gateways update the data:

- *Update COS (Change of state)*: when any data of the input data set change status.
- *Heartbeat interval*: data will be sent automatically according to the configured **Heartbeat rate** in ms.

Note If Heartbeat interval is enabled, a change of state will trigger an immediate update of the data as well, regardless of the set heartbeat rate. I.e. COS is always active.

For both methods the following structure of messages applies.

General telegram structure

The request/response message (e.g. telegram) is structured as shown below:

0	1	...	...	...	...	...	...	...	...	...	...	...	...	n
Command	**Parameter(s)** (content depends on type of command)											**Data**		

Tab. 19: Telegram structure

Parameter	Length	Description
Command	WORD	0h = Undefined (no command) **Polling mode specific** 00F1h = Input data set(s) request message 001Fh = Input data set(s) response message **Auto-update specific** 00E1h = Auto update control 001Eh = Auto update control response 002Eh = Auto update input data set(s) message **Digital outputs read/write** 00F2h = Write output data set settings 002Fh = Response to write output data set settings
Parameter(s)	Length determined by command	As defined in specific command
Data	Length determined by command	As defined in specific command

Error response to invalid messages

The gateway will set the most significant bit of the command word in the event that an invalid or improperly formatted message is received.

Tab. 20: Error response message

Parameter	Length	Description
Command	WORD	Bit 15 of received command will be set (i.e. command of 00F2h would become 80F2h)
Following data	Length determined by command	Unchanged. Returned as it was received

Application requests (Polling) mode

In this mode the gateway will only send any data upon request (e.g. polling). Therefore the application shall send request telegrams as per definition below and the gateway will respond with telegrams structured as per definition below.

Note In order to prevent that the connection will be closed automatically, the application must request data at least every 30 s when Polling mode is active.

Get input data set(s)

The request message is sent by an application to the gateway. The request message telegram shall be structured as shown below:

Tab. 21: Get data set(s) request

Parameter	Length	Value
Command	WORD	00F1h = Data set(s) request message
Request data set 1	WORD	0 = Do not send data set 1 1 = Send data set 1
Request data set 2	WORD	0 = Do not send data set 2 1 = Send data set 2
Request data set 3	WORD	0 = Do not send data set 3 1 = Send data set 3
Request data set 4	WORD	0 = Do not send data set 4 1 = Send data set 4

The response message is returned to the application by the gateway. The response message telegram will be structured as shown below:

Tab. 22: Get data set(s) response

Parameter	Length	Value
Command	WORD	00F1h = Data set(s) response message
Data set 1 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 2 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 3 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 4 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set(s) data	Array of bytes	Data set(s) information

Write output data sets

The following command message is sent by the application to the gateway to write to the output data sets:

Tab. 23: Write output data set setting command

Parameter	Length	Value
Command	WORD	00F2h = Set output data set(s) command message
Output data set 1 length	WORD	0 = Output data set not included in data set(s) data field Non-zero = Length of data set
Output data set 2 length	WORD	0 = Output data set not included in data set(s) data field Non-zero = Length of data set
Output data set 3 length	WORD	0 = Output data set not included in data set(s) data field Non-zero = Length of data set
Output data set 4 length	WORD	0 = Output data set not included in data set(s) data field Non-zero = Length of data set
Output data set 5 length	WORD	0 = Output data set not included in data set(s) data field Non-zero = Length of data set
Data set(s) data	Array of bytes	Data set(s) information

The response message is returned to the application by the gateway. The response message telegram is structured as shown below:

Tab. 24: Write output data set setting response

Parameter	Length	Value
Command	WORD	002Fh = Response to write output data set settings message
Status	WORD	0 = Success. Output data sets written correctly 1 = Error — Can not write output data sets due to either: • Loss of backplane communication • Incorrect routing information

Configuration via Flexi Soft Designer tool

In order to configure the **Application requests (Polling) mode** of the gateway via the Flexi Soft Designer tool, perform the following steps:

- Open the Flexi Soft Designer and load the hardware configuration including the gateway.
- Click on the **Gateways** button above the main window and select the desired gateway or double click the desired gateway in the hardware configuration view to open the gateway configuration dialog.
- Click on **TCP/IP configuration** on the left hand menu. The following dialog appears:

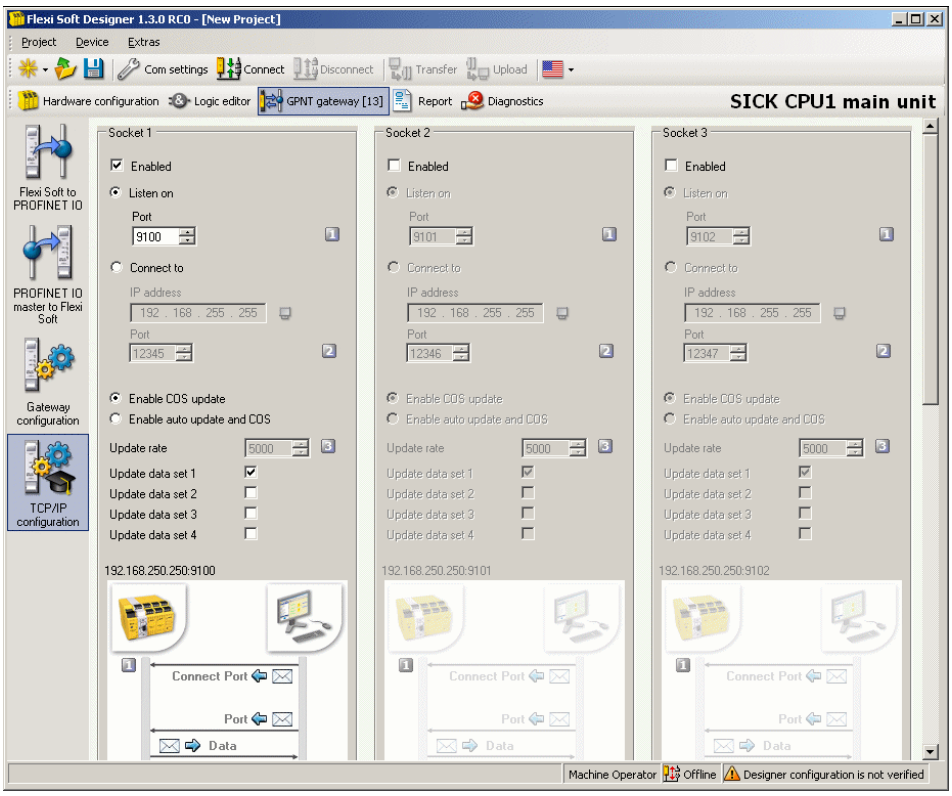


Fig. 17: TCP/IP configuration for Application requests (Polling) mode

- Activate the **Listen on** radio button.
- Enter the **Port** number on which the application will connect.
- Select the update mode: **Enable COS update** or **Enable auto update and COS**.
- If you have selected **Enable auto update and COS**, select the **Update rate** in ms.
- Select which data sets shall be updated: Check the **Update data set n** checkbox.

Gateway writes to Address/Port (Auto update) mode

The gateway can be configured to automatically update the data set information (i.e. the application does not need to send any request messages as it would do in polling mode) once the connection to the application has been made.

The configuration settings are available via the Flexi Soft Designer configuration tool or via the TCP/IP interface itself. Using one interface does not disable the other: The auto update mode could be enabled via Flexi Soft Designer and disabled via TCP/IP command, for example.

Configuration via TCP/IP interface

This command message is sent by an application to the gateway to configure the auto update mode. This message can be used to either disable or enable the auto update mode directly through the TCP/IP interface.

Tab. 25: Auto update mode configuration command

Parameter	Length	Value
Command	WORD	00E1h = Auto update control
Request data set 1	WORD	0 = Do not send data set 1 1 = Send data set 1
Request data set 2	WORD	0 = Do not send data set 2 1 = Send data set 2
Request data set 3	WORD	0 = Do not send data set 3 1 = Send data set 3
Request data set 4	WORD	0 = Do not send data set 4 1 = Send data set 4
Heartbeat mode update rate	WORD	0 = Disable heartbeat messages Non-zero = Enable heartbeat message at specified rate in ms. Minimum = 40 ms

Note Auto update is disabled if all Request Input Data Set flags are set to zero.

The response message returned to the application by the gateway:

Tab. 26: Auto update mode configuration response

Parameter	Length	Value
Command	WORD	001Eh = Response to the auto update control message

Configuration via Flexi Soft Designer tool

In order to configure the **Gateway writes to Address/Port (Auto update)** mode of the gateway via the Flexi Soft Designer tool, perform the following steps:

- Open the Flexi Soft Designer and load the hardware configuration including the gateway.
- Click on the **Gateways** button above the main window and select the desired gateway or double click the desired gateway in the hardware configuration view to open the gateway configuration dialog.
- Click on **TCP/IP configuration** on the left hand menu. The following dialog appears:

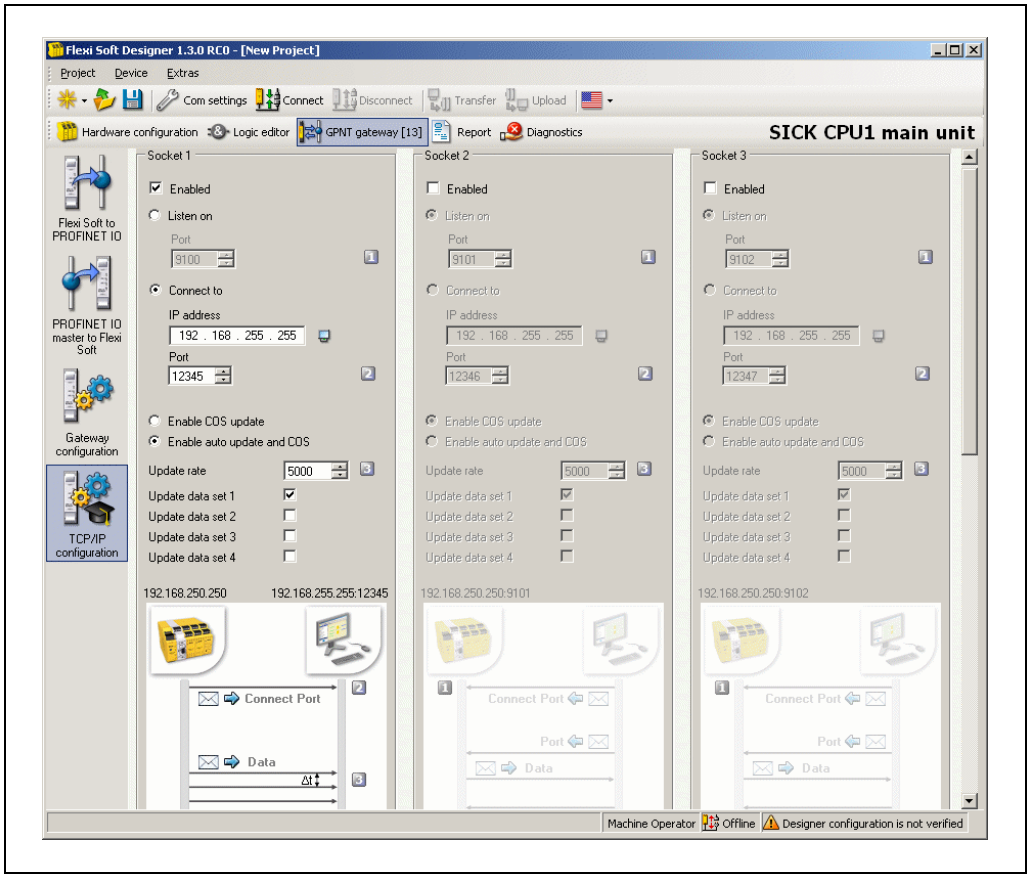


Fig. 18: TCP/IP configuration for auto update

- Activate the **Connect to** radio button.
- Enter the **IP address** and the **Port** number the gateway shall write to.
- Select the update mode: **Enable COS update** or **Enable auto update and COS**.
- If you have selected **Enable auto update and COS**, select the **Update rate** in ms.
- Select which data sets shall be updated: Check the **Update data set n** checkbox.

Normal operation

The following message is sent from the gateway to the application while operating in auto update mode.

Tab. 27: Auto update mode normal operation message

Parameter	Length	Value
Command	WORD	002Eh = Auto update data set(s) message
Data set 1 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 2 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 3 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 4 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set(s) data	Array of bytes (length dependent on set definition)	Data set(s) information. Details see section 3.3 “Data transmitted into the network (input data sets)” on page 14 and chapter 7 “Layout and content of the process image” on page 173.

5.1.3 TCP/IP process image example

The following example shows a possible process image sent by a FX0-GENT gateway via TCP/IP in auto update mode:

Tab. 28: TCP/IP process image example

Byte values [hex]	Part of message	Meaning
00 2E	Command	Auto update data sets (see Tab. 27)
00 32	Command parameters	Length of data set 1: 50 bytes
00 20		Length of data set 2: 32 bytes
00 3C		Length of data set 3: 60 bytes
00 3C		Length of data set 4: 60 bytes
03 FF 03 03	Data set 1 (default byte assignments, see Tab. 5)	Logic results 0-3
C0		Input values module 1: C0 = 11000000 = Inputs I8 and I7 **Active**
03		Input values module 2: 03 = 00000011 = Inputs I2 and I1 **Active**
3F 05 **05 05 00 00** **00 00 00 00**		Input values module 3-12
00 00 00 00 **00 00 00 00** **00 00 00 00**		Output values module 1-12
00 00 00 00 **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00**		Not assigned

Byte values [hex]	Part of message	Meaning
52 A1 10 4C	Data set 2 (see Tab. 5)	Overall CRC (same as system CRC)
52 A1 10 4C		System CRC
00 00 00 00		Reserved
00 00 00 00		
00 00 00 00 **00 00 00 00** **00 00 00 00** **00 00 00 00**		
FF FF FF FF	Data set 3 (see Tab. 5 and section 3.3.6 “Error and status information of the modules” on page 19) Data set 3 data is transferred in Big Endian format, i.e. in 32 bit double word format with the most significant byte placed in the leftmost position.	Status module 0 (FX3-CPUx): OK
FF FF FF FF		Status module 1 (e.g. FX3-XTDI): OK
FD FB FF FF		Status module 2 (e.g. FX3-XTIO): Byte 0: FF = 11111111: No errors Byte 1: FF = 11111111: No errors Byte 2: FB = 11111011: Input 3 external test signal failure. Byte 3: FD = 11111101: Output 1 stuck-at-low error
FF FF FF FF **FF FF FF FF** **FF FF FF FF** **FF FF FF FF**		Status modules 3-6: OK
FF FF FF FF **FF FF FF FF** **FF FF FF FF** **FF FF FF FF** **FF FF FF FF** **FF FF FF FF**		Status modules 7-12 (no modules present)
FF FF FF FF		Status module 13 (e.g. FX0-GENT): OK
FF FF FF FF		Status module 14 (no module present)
00 00 00 00 **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00** **00 00 00 00**	Data set 4	Reserved

5.1.4 TCP/IP socket monitor

The TCP/IP socket monitor allows you to view the input data sets that a gateway transfers into the network via TCP/IP, issue commands to the gateway and to write data to the gateway's output data sets.

The TCP/IP socket monitor is a separate program that is installed with the Flexi Soft Designer. You will find the TCP/IP socket monitor in the Windows Start menu under "Programs/SICK/Flexi Soft Designer/Tools/TCPIP socket monitor".

In order to connect to the gateway via the socket monitor, at least one TCP/IP socket must be enabled on the gateway (see section 5.1.2 "Ethernet TCP/IP socket interface" on page 37).

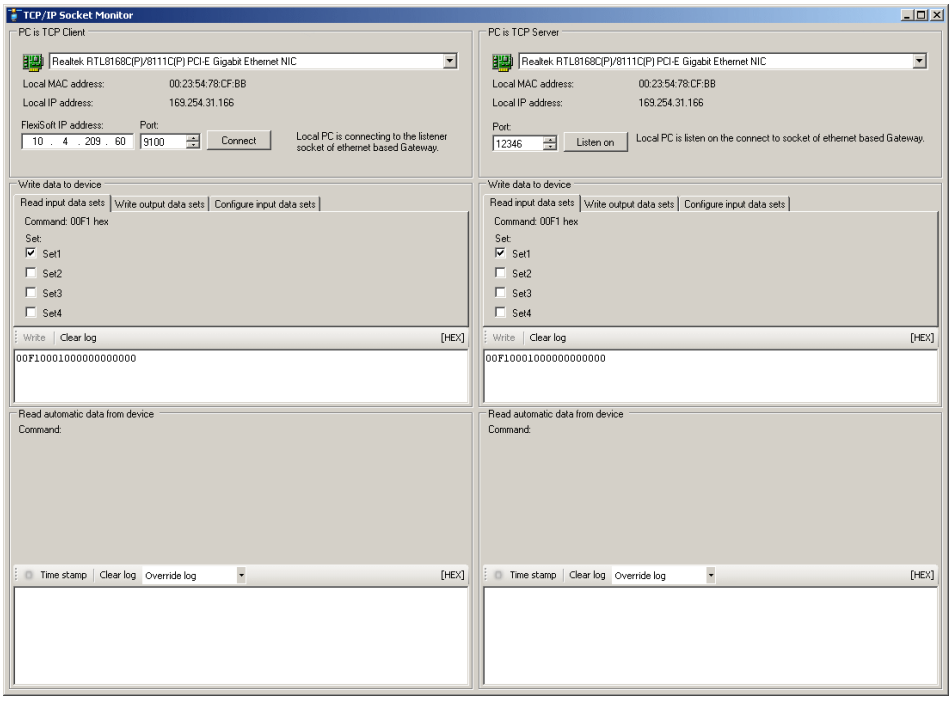
How to use the socket monitor:

- Power up the Flexi Soft system.
- Connect one of the two Ethernet ports of the gateway with your Ethernet network using a shielded Ethernet cable.
- Connect a PC (or notebook) to the same Ethernet network. Ensure the IP address settings of the gateway and the PC match the network setup.

Note You can also connect your PC directly to one of the two Ethernet ports of the gateway. In this case, you can either adapt the IP address settings of your PC or the IP address settings of the gateway to match the other device's IP setup.

- Open the TCP/IP socket monitor. The following window opens:

Fig. 19: Socket monitor window



The TCP/IP socket monitor can connect to the gateway either as TCP client or as TCP server, depending on the socket configuration of the gateway.

PC is TCP Client

If at least one socket is enabled and if the gateway is configured to **Listen** on this socket, the PC can connect to the gateway as client.

In a dropdown list in the **PC is TCP Client** area, all available network adapters are displayed:

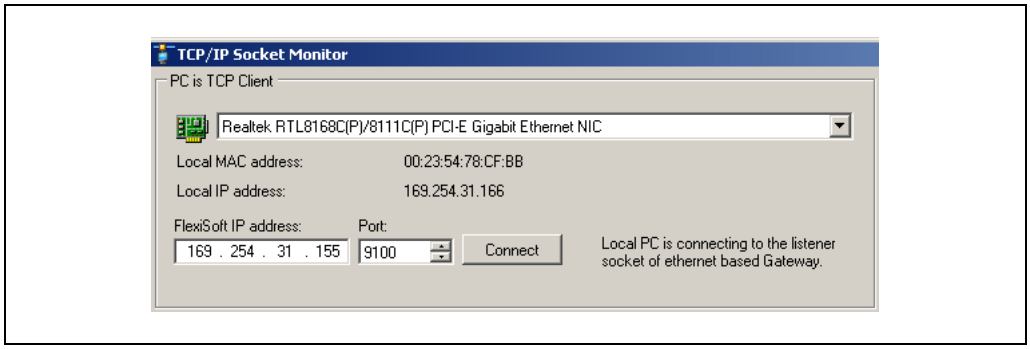


Fig. 20: PC is TCP Client area — Gateway is configured to listen

- Select the network adapter that is connected to the gateway. The MAC address and the IP address of the selected network adapter are displayed beneath the dropdown list.
- Enter the **Flexi Soft IP address** of the gateway and the **Port** number of the enabled socket.
- Click on **Connect** to establish the communication to the gateway.

PC is TCP Server

If at least one socket is enabled and if the gateway is configured to **Connect** on this socket, the PC can connect to the gateway as server.

In a dropdown list in the **PC is TCP Client** area, all available network adapters are displayed:

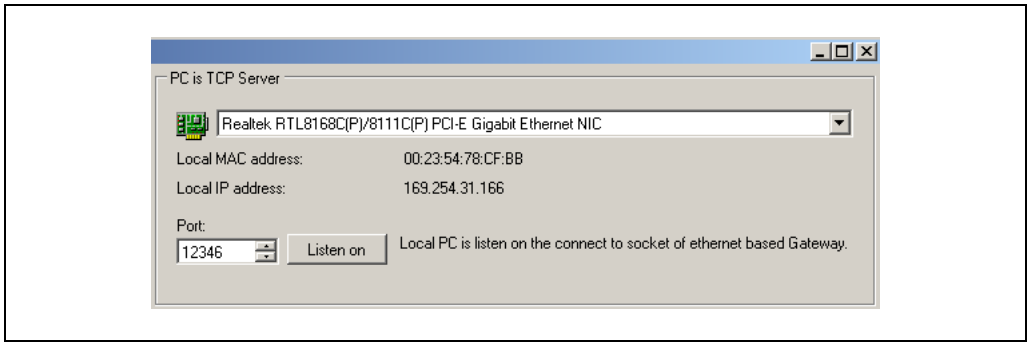


Fig. 21: PC is TCP Server area — Gateway is configured to connect

- Select the network adapter that is connected to the gateway. The MAC address and the IP address of the selected network adapter are displayed beneath the dropdown list.

Note The gateway socket must be configured to connect to the local IP address of the PC.

- Enter the **Port** number on which the enabled socket is configured to connect.
- Click on **Listen on** to establish the communication to the gateway.

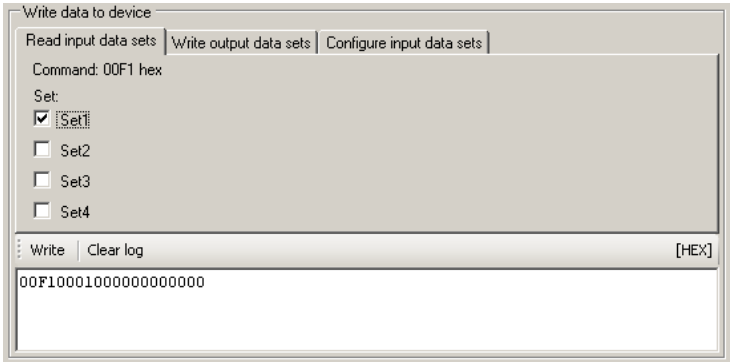
Once the connection to the gateway is established, the further proceedings are alike, no matter whether the PC is connected as client or as server. It is possible to establish a client and a server connection at the same time, if the gateway socket configuration allows this.

Write data to device — How to control the gateway

The **Write data to device** area consists of three file cards which are used to assemble different commands that can be sent to the gateway. The commands are assembled automatically as you activate or deactivate the different options on the file cards and are displayed in the log window. A click on the **Write** button sends the displayed command to the gateway. The **Clear log** button deletes the displayed command from the log window. It is also possible to edit the command in the log window manually.

Under the **Read input data sets** file card you can request input data sets from the gateway using the 00F1h command (see Tab. 21 on page 41).

Fig. 22: Write data to device — Read input data sets

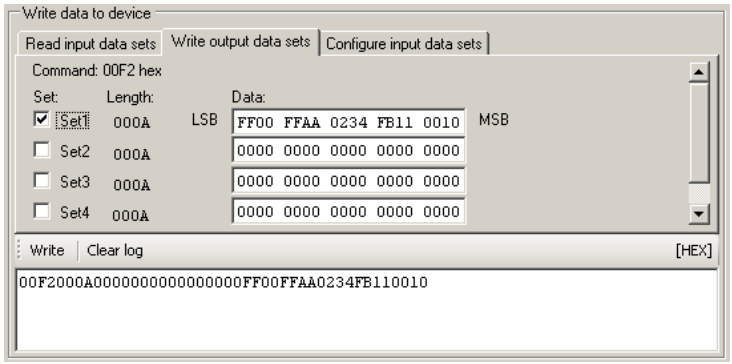


- Activate all data sets that you want to receive. The command that will be sent is shown in the log window.
- Click on the **Write** button above the log window to send the command to the gateway. The gateway will respond with a 00F1h response message (see Tab. 22 on page 41) that will be shown in the **Read automatic data from device** area below.

Note If the gateway is configured to send data sets on this socket cyclically or on change of state (COS), the gateway's response may be overwritten very soon by the next message from the gateway. In this case you should modify the configuration under the **Configure input data sets** file card (see below).

Under the **Write output data sets** file card you can write data to the gateway's output data sets using the 00F2h command (see Tab. 23 on page 42).

Fig. 23: Write data to device — Write output data sets

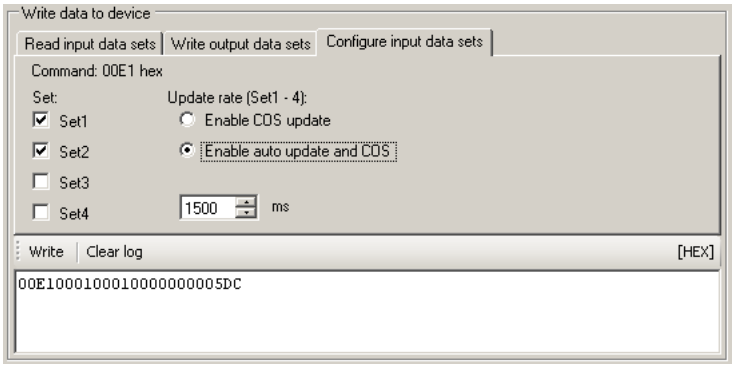


- Activate all data sets that you want to send and edit the data in the input field for the respective data set. The command that will be sent is shown in the log window below.

➢ Click on the **Write** button above the log window to send the command to the gateway. The gateway will respond with a 00F2h response message to indicate whether the output data sets have been written correctly or if an error has occured (see Tab. 24 on page 42). The result will be shown in the **Read automatic data from device** area below.

Under the **Configure input data sets** file card you can configure the gateway to send input data sets either on change of state (COS) or cyclically (auto update) using the 00E1h command (see Tab. 25 on page 44).

Fig. 24: Write data to device — Configure input data sets



➢ Activate all data sets that you want to receive, choose the update mode (i.e. either COS or auto update and COS) and enter the **Update rate** (40 to 65535 ms), if auto update is desired. The command that will be sent is shown in the log window.

➢ Click on **Write** to send the command to the gateway. The gateway will respond with a 00E1h response message (see Tab. 26 on page 44) that will be shown in the **Read data from device automatically** area below.

➢ Depending on the configuration, the gateway will follow up with 00E2h messages, i.e. it will send the activated data sets as on COS and cyclically as configured (see Tab. 27 on page 46).

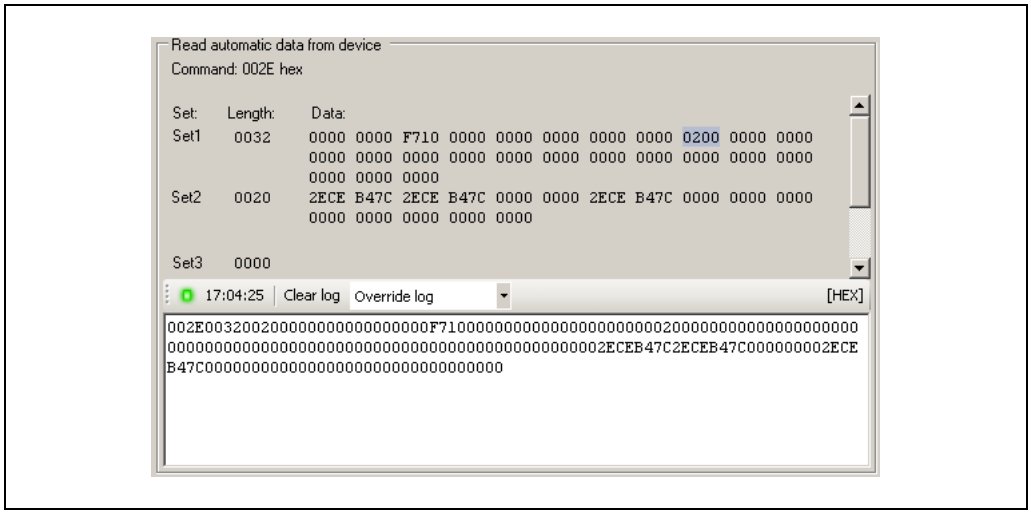
Notes

- The initial behaviour of the gateway when you open the socket monitor depends on the gateway's configuration for the respective socket in your project or on the last 00E1h command the gateway has received and does not correspond necessarily with the settings displayed in the **Configure input data sets** file card.
- The changes you make to the input data sets configuration here will not be stored in your project but will change the behaviour of the gateway temporarily, i.e. until it receives another 00E1h command or until the Flexi Soft system is restarted.

Read automatic data from device — Monitoring the gateway's messages

The **Read automatic data from device** area shows the messages sent by the gateway.

Fig. 25: Read automatic data from device



- The last **Command** and the **Data sets** that have been sent by the gateway are displayed in the grey part of the area. The data are displayed in blocks of four bytes. Data blocks that have changed value since the last message has been received are highlighted blue.
- The log window shows the complete command string the gateway has sent.
- The communication indicator in the upper left corner of the log window title bar lights up green if currently data are received from the gateway.
- The time stamp in the log window title bar indicates when the last command string has been received.
- The **Clear log** button erases the log.
- You can choose from the dropdown list whether the next received command string shall override the last one or if it shall be appended at the top of the log.

5.2 EtherNet/IP gateway

The following Flexi Soft gateway can be used for EtherNet/IP: FX0-GENT.

You will find the EDS files and the device icon for PLC interfacing ...

- in the Internet on the FX0-GENT product page on www.sick.com.
- in the Flexi Soft Designer program folder on your hard disk (default installation folder is "C:\programs\SICK\FlexiSoft\DeviceDescriptions\...").

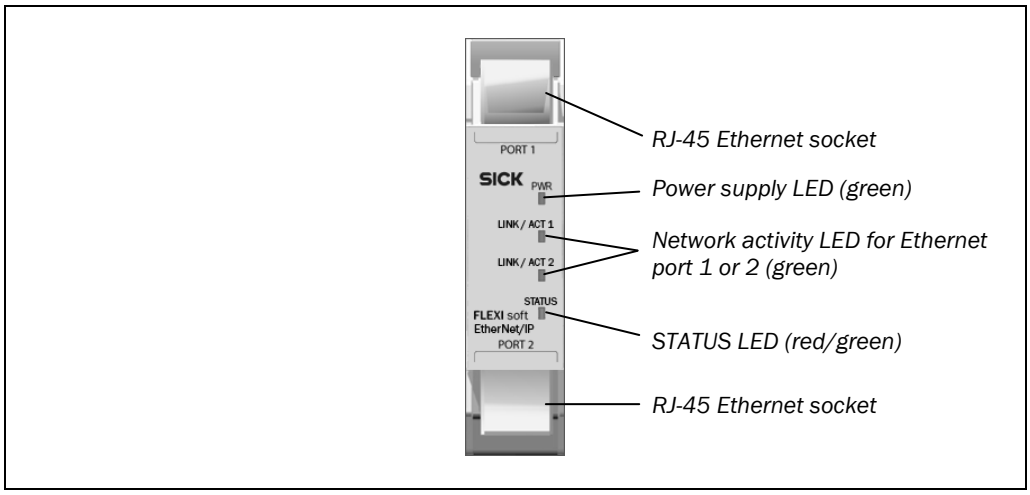
Flexi Soft EtherNet/IP gateways with firmware V1.xx.x support only EtherNet/IP explicit messaging (Class 3). Implicit messaging (Class 1) is not supported. Please use the EDS file **SICK_FX0_GENT_1.00.eds** for these gateways.

Flexi Soft EtherNet/IP gateways with firmware ≥ V2.00.0 support EtherNet/IP implicit messaging and explicit messaging (Class 1 and Class 3). Please use the EDS file **SICK_FX0_GENT_2.02.eds** for these gateways.

5.2.1 Interfaces and operation

The FX0-GENT is equipped with an integrated three-port switch for connection to the Ethernet network. Two RJ-45 sockets are available for the connection. The switch functionality allows the FX0-GENT to be used for connection to another Ethernet component (e.g. connection to a notebook) without having to interrupt the Ethernet connection to the network.

Fig. 26: Interfaces and display elements of the FX0-GENT



Tab. 29: Meaning of the LED displays of the FX0-GENT

Symbol description:
○: LED is off
● Green: LED lights up green
☼● Red: LED flashes red

LED		Meaning
PWR	○	No power supply
	● Green	Power supply switched on
LINK/ACT 1 LINK/ACT 2	○	No Ethernet connection
	● Green	Ethernet connection active, no data transmission
	☼● Green	Ethernet connection active, data transmission
STATUS[6)]	○	Power-up
	● Green	Executing (live process data from/to FX3-CPUx)
	☼● Green	1 Hz: Idle
	☼● Red	1 Hz: Configuring/configuration required 2 Hz: Critical fault on gateway
	● Red	Critical fault on another module
	☼● Red/green	Executing, but faulty or no Ethernet communication

Note Error elimination is described in section 5.2.10 “Diagnostics and troubleshooting” on page 74.

Power-up sequence

On power-up, the following LED test sequence is performed:

- STATUS LED ○ Off for 6 s
- STATUS LED ● Red for 0.25 s
- STATUS LED ● Green for 0.25 s
- STATUS LED ○ Off

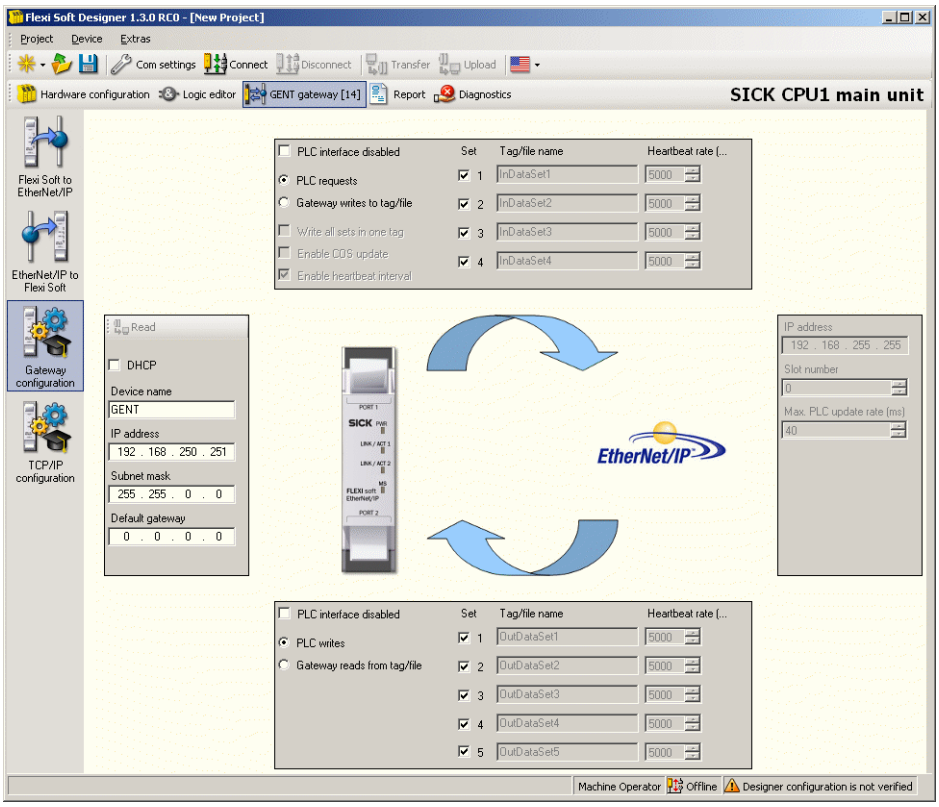
[6)] On older versions of the FX0-GENT, the STATUS LED is called MS LED.

5.2.2 Basic configuration — Assigning a device name and IP address

Configuration of the FX0-GENT is performed via the Flexi Soft Designer tool.

- Open the Flexi Soft Designer and load the hardware configuration including the EtherNet/IP gateway.
- Click on the **Gateways** button above the main window and select the FX0-GENT or double click the FX0-GENT in the hardware configuration to open the gateway configuration dialog.
- Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 27: EtherNet/IP gateway configuration dialog



- If desired, change the **Device name** for the Flexi Soft gateway.
- Enter a valid **IP address** for the Flexi Soft gateway, and if required a valid **Subnet mask** and a valid IP address for a **Default gateway**.
- Click **Connect** to go online and download the configuration to the Flexi Soft system.

5.2.3 EtherNet/IP Class 1 communication — Implicit messaging

Note

- Only Flexi Soft EtherNet/IP gateways with firmware ≥ V2.00.0 support both EtherNet/IP implicit messaging and explicit messaging (Class 1 and Class 3). If your gateway has an older firmware version, please refer to section 5.2.6 “Ethernet/IP Class 3 communication — Explicit messaging” on page 60. Please use the EDS file **SICK_FX0_GENT_1.00.eds** for these older gateways.

General overview

Implicit messaging is a method of communication between EtherNet/IP PLCs and devices.

- Implicit messaging uses Ethernet UDP messages.
- Implicit messaging is cyclic in nature. Input and output data is exchanged between the controllers and devices at regular time intervals.
- Message delivery is not guaranteed.
- Multicast addressing is possible.

Class 1 functionality

To establish Class 1 communication, a connection between the PLC and the EtherNet/IP gateway must be created. To this end, the PLC must send a Forward Open message to the gateway. The gateway then checks the received parameters, responds with a success or fail status message (Forward Open response) and, if successful, includes a set of connection parameters.

The Forward Open message from the PLC to the EtherNet/IP gateway includes the following parameters:

- input connection type (gateway to PLC: either point-to-point or multicast)
- input assembly object instance number
- input data length
- output connection type (PLC to gateway: only point-to-point is accepted)
- output assembly object instance number (unused if input data only)
- output data length (unused if input data only)
- requested packet interval

The EtherNet/IP gateway then returns the following parameters in the Forward Open response:

- status of connection attempt
  - SUCCESS: If the received parameters are acceptable and the gateway has sufficient bandwidth and memory resources, the connection will be accepted. The STATUS LED is ● Green.
  - FAILURE: If either the received parameters are incorrect or the gateway does not have sufficient bandwidth and memory resources, the connection will be rejected. The STATUS LED blinks ☼ Red/green.
- The IP address and UDP socket port number on which the PLC must listen for input messages:
  - If the input connection is point-to-point, this is the PLC’s IP address.
  - If the input connection is multicast, the gateway sends the multicast address on which the PLC must listen for input messages.
- The packet interval that will be allowed by the gateway. This may be the same as or greater than the interval requested by the PLC.

Once a connection has been successfully created, data can then be exchanged between the PLC and the EtherNet/IP gateway.

The connection will remain open until it is closed by either the PLC or the EtherNet/IP gateway.

Packet update interval

The packet update interval for Class 1 connections that will be returned to the EtherNet/IP PLC in the Forward Open response depends on the following factors:

- the value for the **Requested Packet Interval** received from the EtherNet/IP PLC in the Forward Open message
- the **Maximum PLC Update Rate** as configured in the **Gateway configuration** dialog of the Flexi Soft Designer
- the 10 ms system clock that the EtherNet/IP gateway operates on

If the Requested Packet Interval is less than the Maximum PLC Update Rate, the packet update interval will be set to the Maximum PLC Update Rate. Otherwise, it will be set to the Requested Packet Interval.

If the packet update interval is not a multiple of 10 ms (10, 20, 30, 40, etc.), then the packet update interval will be adjusted up to the next multiple of 10 ms.

Tab. 30: Examples for the packet update interval

Requested Packet Interval	Maximum PLC Update Rate	Actual packet update interval	Description
5 ms	10 ms	10 ms	Set to Maximum PLC Update Rate
10 ms	10 ms	10 ms	Requested Packet Interval accepted
15 ms	20 ms	20 ms	Set to Maximum PLC Update Rate
15 ms	10 ms	20 ms	Requested Packet Interval adjusted upward to 20 ms
20 ms	25 ms	30 ms	Maximum PLC Update Rate adjusted upward to 30 ms
40 ms	30 ms	40 ms	Requested Packet Interval accepted
32 ms	30 ms	40 ms	Requested Packet Interval adjusted upward to 40 ms
48 ms	40 ms	50 ms	Requested Packet Interval adjusted upward to 50 ms
50 ms	40 ms	50 ms	Requested Packet Interval accepted

Bandwidth limitations

The maximum number of Class 1 messages per second is limited by the FX3-CPUx. At 50 % of available CPU bandwidth, this is approximately 200 messages per second or one Class 1 connection at 10 ms I/O update rate (the system clock frequency of the FX0-GENT is 10 ms).

Tab. 31: Recommended bandwidths for Class 1 messages

PLC update rate [ms]	Cyclic two-way I/O connections	Cyclic input-only multicast connections
10	1	2
20	2	4
40	Up to 4	Up to 8

Note The gateway will not enforce these bandwidth recommendations. However, if the bandwidth used for Class 1 communication exceeds 200 messages per second, the RS-232 interface and the Ethernet TCP/IP interface will slow down.

Point-to-point and multicast connections

For EtherNet/IP PLC to gateway:

- Only point-to-point connections will be accepted.
- Multicast connections will not be accepted.

For gateway to EtherNet/IP PLC:

- Both point-to-point and multicast connections will be accepted.
- Multicast connections can be either exclusive owner or multiple listeners.

Maximum number of Class 1 connections

For I/O connections involving *both input and output* data sets:

- Each output data set can be controlled by only one I/O connection.
- I/O connection attempts to control an output data set that is already controlled by another I/O connection will be rejected.
- If each I/O connection controls only one output data set, then up to 5 I/O connections can be active simultaneously.
- The maximum number of possible I/O connections decreases if an I/O connection controls more than one output data set.

For connections involving *only input* data sets (gateway to PLC):

- Depending on the gateway's bandwidth, up to 32 connections may be created at one time if all request the same set of input data.
- The maximum number of connections requesting different data that can be supported depends on the requested data rates and the available gateway bandwidth. If the connections require more bandwidth than available, the gateway will slow down and not be able to return the input data at the requested data rates.

Class 1 data access — Input data sets

- All four input data sets are included in one array that can be read by all Class 1 PLCs.
- The start of the input data received by the PLC is defined by the assembly instance number. Each input instance number corresponds to the start of an input data set.
- The length determines how much input data is received by the PLC. This allows partial and/or multiple input data sets to be received by the PLC. For example, the first 20 bytes of input data set 1 or all input data sets could be read by the PLC.
- All input data sent to the PLC must be sequential. This means that input data sets 1 and 2 or input data sets 1, 2, and 3 can be sent together. However, input data sets 1 and 3 can not be sent together because they are not sequential.

Tab. 32: Class 1 read access points to input data sets

Assembly instance	Byte index	Length [bytes]	Input data set	Description	Valid read lengths [bytes]
1	0-49	50	1	Starting at input data set 1 Can read input data sets 1-4	1-202
2	50-81	32	2	Starting at input data set 2 Can read input data sets 2-4	1-152
3	82-141	60	3	Starting at input data set 3 Can read input data sets 3-4	1-120
4	142-201	60	4	Can read input data set 4	1-60

Class 1 data access — Output data sets

- All five output data sets are included in one array that can be written by all Class 1 PLCs.
- The start of the output data is defined by the assembly instance number. Each output instance number corresponds to the start of an output data set.

- The length determines how much output data is sent by the PLC. This allows a PLC to write to one or multiple output data sets. For example, output data set 1 or all five output data sets could be written.
- Because partial output data sets cannot be written, the output length must be a multiple of 10 bytes. The length must be 10 to write one output data set, 20 for two data sets, and so on.
- All output data written by the PLC must be sequential. This means that e.g. output data sets 1 and 2 or output data sets 1, 2, and 3 can be written together. However, output data sets 1 and 3 can not be written together because they are not sequential.

Tab. 33: Class 1 write access points to output data sets

Assembly instance	Byte index	Length [bytes]	Output data set	Description	Valid write lengths [bytes]
5	0-9	10	1	Starting at output data set 1 Can set output data sets 1-5	10 = Output data set 1 20 = Output data sets 1+2 30 = Output data sets 1-3 40 = Output data sets 1-4 50 = Output data sets 1-5
6	10-19	10	2	Starting at output data set 2 Can set output data sets 2-5	10 = Output data set 2 20 = Output data sets 2+3 30 = Output data sets 2-4 40 = Output data sets 2-5
7	20-29	10	3	Starting at output data set 3 Can set output data sets 3-5	10 = Output data set 3 20 = Output data sets 3+4 30 = Output data sets 3-5
8	30-39	10	4	Starting at output data set 4 Can set output data sets 4+5	10 = Output data set 4 20 = Output data sets 4+5
9	40-49	10	5	Starting at output data set 5 Can set output data set 5	10 = Output data set 5

Assembly object definition

All Class 1 data must be transferred using the assembly object. The assembly object interface is used to directly tie Vendor Specific objects to a standard interface which the EtherNet/IP PLC uses to communicate with the device.

For the Flexi Soft EtherNet/IP gateway, the assembly object corresponds to the Full Data Set Transfer object (72h), which provides access to the input and output data sets (see page 65). Each instance of the assembly object corresponds to one or more Full Data Set Transfer object attributes.

The assembly object defines the interface by which a Class 1 PLC can ...

- request the input data set information from the Flexi Soft gateway.
- write the output data set information to the Flexi Soft gateway.

Tab. 34: Assembly object class attributes

Attribute ID	Name	Data type	Data values	Access rule
1	Revision	UINT	1	Get
2	Max. instance	UINT	9	Get
3	Num. instances	UINT	9	Get

Tab. 35: Assembly object instance definitions

Assembly instance no.	Description	Data type	Data values	Access rule	Corresponding Full Data Transfer object attributes
Flexi Soft to Network					
1	Request input data sets 1 to 4 data	BYTE[202] Valid read lengths: 1-202	0-255	Get	1, 2, 3, 4
2	Request input data sets 2 to 4 data	BYTE[152] Valid read lengths: 1-152	0-255	Get	2, 3, 4
3	Request input data set 3 and 4 data	BYTE[120] Valid read lengths: 1-120	0-255	Get	3, 4
4	Request input data set 4 data	BYTE[60] Valid read lengths: 1-60	0-255	Get	4
Network to Flexi Soft					
5	Write output data set 1 to 5 data	BYTE[50] Valid write lengths: 10 = Set 1 20 = Sets 1-2 30 = Sets 1-3 40 = Sets 1-4 50 = Sets 1-5	0-255	Get/Set	5, 6, 7, 8, 9
6	Write output data sets 2 to 5 data	BYTE[40] Valid write lengths: 10 = Set 2 20 = Sets 2-3 30 = Sets 2-4 40 = Sets 2-5	0-255	Get/Set	6, 7, 8, 9
7	Write output data sets 3 to 5 data	BYTE[30] Valid write lengths: 10 = Set 3 20 = Sets 3-4 30 = Sets 3-5	0-255	Get/Set	7, 8, 9
8	Write output data sets 4 and 5 data	BYTE[20] Valid write lengths: 10 = Set 4 20 = Sets 4-5	0-255	Get/Set	8, 9
9	Write output data set 5 data	BYTE[10] Valid write lengths: 10 = Set 5	0-255	Get/Set	9

Tab. 36: Assembly object instance attributes

Attribute ID	Name	Data type	Data values	Access rule
3	Data	Array of BYTE	0-255	Get/Set
4	Data length	UINT	Maximum number of bytes in attribute 3	Get

Attribute 3 — Request/Write Data: either the readable input data or the writeable output data, dependent on the instance number

Attribute 4 — Data length: maximum data length for each assembly instance

Common services

Tab. 37: Assembly object common services

Service code	Implemented in class	Implemented in instance	Service name
01h	Yes	No	Get_Attributes_All
0Eh	Yes	Yes	Get_Attribute_Single
10h	No	Yes	Set_Attribute_Single
02h	No	No	Set_Attribute_All

5.2.4 Example configuration of implicit messaging with Rockwell RSLogix 5000

You will find a description of the configuration of a class 1 connection using Rockwell RSLogix 5000 in the brochure “Flexi Soft Ethernet IP: Implicit Messaging with Rockwell RSLogix 5000” (SICK part no. 8015359). This brochure is available for download in PDF format at www.sick.com.

5.2.5 Example configuration of implicit messaging with an OMRON PLC

You will find a description of the configuration of a class 1 connection using an OMRON PLC in the brochure “Flexi Soft Ethernet IP: Implicit Messaging with an Omron PLC” (SICK part no. 8015341). This brochure is available for download in PDF format at www.sick.com.

5.2.6 Ethernet/IP Class 3 communication — Explicit messaging

General overview

Explicit messaging is a method of communication between EtherNet/IP PLCs and devices.

- Explicit messaging uses Ethernet TCP/IP messages.
- Explicit messaging is not cyclic in nature. The PLC and the devices must send individual messages to each other.
- Message delivery is guaranteed.
- Multicast addressing is not possible.

Transfer methods

The configuration steps in this section specify how the data to the higher-level PLC are transferred. In general, there are two different transfer *methods* available for both transfer *directions* such as Flexi Soft to Network and Network to Flexi Soft:

- **Gateway writes to tag/file** and/or **Gateway reads from tag/file** — the FX0-GENT gateway operates as *master*. It writes the data into and/or reads from the PLC memory.
- **PLC requests** and/or **PLC writes** — the FX0-GENT gateway operates as *slave*. The PLC requests the data from the gateway and/or writes the data to the gateway.

Both methods can be mixed. E.g. it is possible to configure the gateway as master for the Flexi Soft to Network direction (option **Gateway writes to tag/file** activated) while it operates at the same time as slave for the Network to Flexi Soft direction (option **PLC writes** activated).

Number of possible connections

The number of possible connections to the PLC depends on whether the FX0-GENT is operated as *master* or *slave*. Depending on the setting, up to 128 PLCs can address the FX0-GENT at the same time.

Tab. 38: Number of possible connections

Operating mode of the FX0-GENT	Maximum connections
Rx (To PLC) Transfer mode: Gateway writes to Tag/File Tx (From PLC) Transfer mode: Gateway reads from Tag/File	Rx and Tx: 1
Rx (To PLC) Transfer mode: Gateway writes to Tag/File Tx (From PLC) Transfer mode: PLC writes	Rx: 1 Tx: 127
Rx (To PLC) Transfer mode: PLC requests Tx (From PLC) Transfer mode: Gateway reads from Tag/File	Rx: 127 Tx: 1
Rx (To PLC) Transfer mode: PLC requests Tx (From PLC) Transfer mode: PLC writes	Rx and Tx: 128

Configuration process

The following table outlines the configuration process depending on the transfer method:

Tab. 39: Configuration guideline — Gateway as master

Gateway is master (Gateway writes to Tag/File and/or Gateway reads from Tag/File)	
To do in the gateway configuration (via Flexi Soft Designer tool)	**To do in the PLC program and/or EtherNet/IP network configuration tool**
Select which data shall be written to/read from the PLC	–
Define where in the PLC memory the selected data shall be written to: Enter tag names. Example: InDataSet1 And/or define where in the PLC memory the selected data shall be read from: Enter tag names. Example: OutDataSet1	Define exactly the same tag names in the PLC program. Example: InDataSet1 INT[25] OutDataSet1 INT[5] The data type shall be INT.
Select how often this data shall be transmitted.	–
Define where the data shall be read from/written to in the EtherNet/IP network: Enter the IP address and controller slot number of the PLC.	–

Tab. 40: Configuration guideline — Gateway as slave

Gateway is slave (PLC requests and/or PLC writes)	
To do in the gateway configuration (via Flexi Soft Designer tool)	**To do in the PLC program and/or EtherNet/IP network configuration tool**
–	Download and install the FX0-GENT EDS file from www.sick.com.
–	Integrate the FX0-GENT into the EtherNet/IP network via network configuration tool (i.e. RSNetworx).
–	Program the explicit message “Get_Attribute_...” or “Set_Attribute_...” in the PLC program to read/write data from/to the gateway
–	Program the trigger for sending the explicit messages.

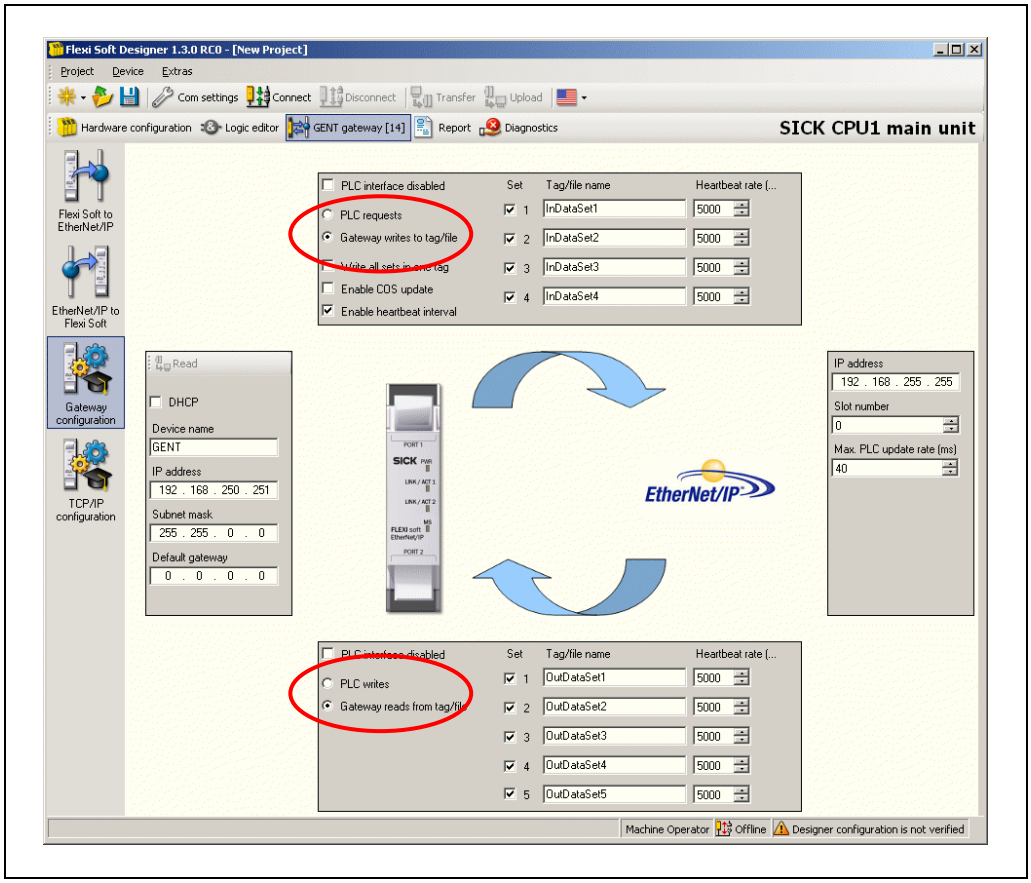
Method 1: Gateway writes to/reads from Tag/File — FX0-GENT writes the data into/reads the data from the PLC memory

In this operating mode the FX0-GENT as a *master* writes the data of all activated data sets into the specified memory areas of the PLC. The only task for the PLC programmer is to define a controller tag name which matches the gateway configuration tag name.

In order to configure the gateway to be *master*, perform the following steps:

- Open the Flexi Soft Designer and load the hardware configuration including the EtherNet/IP gateway.
- Click on the **Gateways** button above the main window and select the FX0-GENT or double click the FX0-GENT in the hardware configuration view to open the gateway configuration dialog.
- Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 28: EtherNet/IP gateway configuration as master



- Within the **Gateway configuration** dialog, select the transfer method by activating **Gateway writes to tag/file** for the Flexi Soft to Network direction and **Gateway reads from tag/file** for the Network to Flexi Soft direction.
- Select which data shall be written to/read from the PLC by checking the checkbox for the required data set.
- Define where in the PLC memory the selected data shall be written to or read from: Enter tag names into the **Tag/file name** edit fields (max. 20 characters).
- Select **Write all sets in one tag** if all data sets shall be written into one tag in the PLC memory. In this case, the tag defined for data set 1 will be used.

➢ Define how often the data shall be transmitted to the PLC:
  – Select **Enable COS update** (update on change of state) if the FX0-GENT is to update the data in the PLC immediately when changes occur in the data sets.
  – Select **Enable heartbeat interval** to activate updating of the selected data sets with the set **Heartbeat rate** in ms.
  – Both options may be selected at the same time.
➢ Define how often the data shall be read from the PLC:
  – Enter a **Heartbeat rate** in ms to activate updating of the selected data sets with the set time.
➢ Define where the data shall be read from/written to in the EtherNet/IP network: Enter the **IP address** and controller **Slot number** of the PLC.

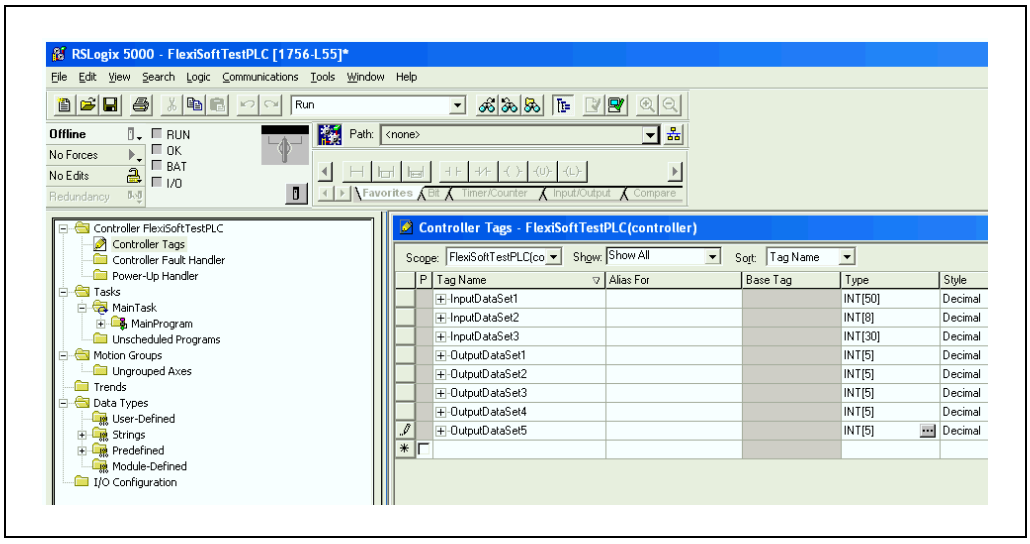
Note The configuration is considered faulty, if the PLC IP address is zero and either **Gateway writes to tag/file** for the Flexi Soft to Network direction and/or **Gateway reads from tag/file** for the Network to Flexi Soft direction is activated.

➢ The **Max. PLC update rate (ms)** defines the maximum rate (the minimum time interval) for transferring the data sets to and from the PLC. Settings occur dependent on the PLC processing speed. Minimum = 10 ms, maximum = 65,535 ms. The default value of 40 ms is suitable for most PLCs.

Notes
  – If the value entered for the **Max. PLC update rate** is greater than the **Heartbeat rate** set for writing to or reading from the PLC, the heartbeat rate will be automatically increased (i.e. slowed down) to this value.
  – All data sets are transferred to the PLC in 16 bit integer format with the first byte placed in the most significant, or leftmost byte of the integer.

➢ Click **Connect** to go online and download the configuration to the Flexi Soft system.
➢ Open the PLC programming tool.
➢ Define the PLC tag names as previously configured in the Flexi Soft EtherNet/IP gateway. Fig. 29 shows an example for the definition of tag names in a PLC program written with RSLogix:

Fig. 29: Example of tag names in a PLC program



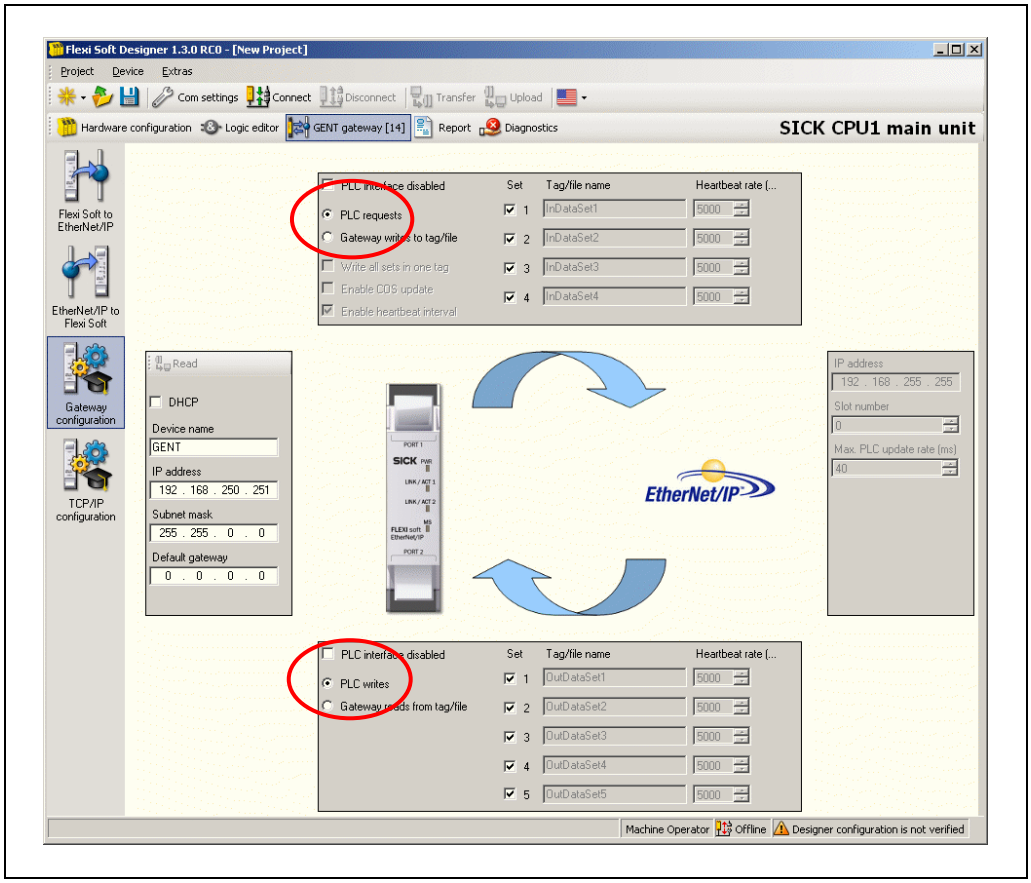
Notes
- Tag names for Allen Bradley SLC/PLC-5 PLCs must begin with a “$” (i.e. $N10:0).
- Tag names for Allen Bradley MicroLogix PLCs must begin with a “#” (i.e. #N10:0).

Method 2: Polling mode — PLC requests the data from/PLC writes the data to the FX0-GENT

In this operating mode the FX0-GENT operates as *slave*. It sends the data to the PLC upon request and the PLC writes the data to the gateway. If this operating mode is desired:

- Open the Flexi Soft Designer and load the hardware configuration including the EtherNet/IP gateway.
- Click on the **Gateways** button above the main window and select the FX0-GENT or double click the FX0-GENT in the hardware configuration view to open the gateway configuration dialog.
- Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 30: EtherNet/IP gateway configuration as slave



- Within the **Gateway configuration** dialog, select the transfer method by activating **PLC requests** for the Flexi Soft to Network direction, **PLC writes** for the Network to Flexi Soft direction.
- Select which data shall be requested or written by the PLC by checking the checkboxes for the required data sets.
- Click **Connect** to go online and download the configuration to the Flexi Soft system.
- Program the explicit messaging in the PLC.

Polling data sets via explicit messaging

The FX0-GENT supports two vendor specific objects which can be polled via explicit messaging:

- The *Full Data Set Transfer* object allows to poll for each of the data sets. There is one instance where each attribute represents a data set.

- The *Individual Input Data Set Transfer* object allows to poll for the individual data set parameters. There is one instance per data set and each attribute represents one byte of the data set.

Full Data Set Transfer object definition (72h — one instance)

The vendor specific **Full Data Set Transfer** object defines the attributes by which the PLC can:

- request the complete input data set information from the FX0-GENT.
- write the complete output data set information to the FX0-GENT.

Class attribute (instance 0)

Tab. 41: Full Data Set Transfer object (72h) class attribute (instance 0)

Attribute ID	Name	Data type	Data values	Access rule
1	Revision	UINT	1	Get
2	Max. instance	UINT	1	Get
3	Num. instances	UINT	1	Get

Instance attribute (instance 1)

These attributes provide access to input and output data sets. *Get Attribute Single* requests for a specific input data set will return the input data set information. *Get Attribute All* requests will return all enabled input data sets.

All data set information will be returned in integer (16 bit word) format. For byte data, the first byte will be placed in the most significant or leftmost byte of the integer and the second byte will be placed in the least significant or rightmost byte of the integer.

Tab. 42: Full Data Set Transfer object (72h) instance attribute (instance 1)

Attribute ID	Name	Data type	Data values	Access rule
Flexi Soft to Network				
1	Request input data set 1 specific data	Array of UINT	0-255	Get
2	Request input data set 2 specific data	Array of UINT	0-255	Get
3	Request input data set 3 specific data	Array of UINT	0-255	Get
4	Request input data set 4 specific data	Array of UINT	0-255	Get
Network to Flexi Soft				
5	Write the output data set 1 specific data	Array of UINT	0-255	Set
6	Write the output data set 2 specific data	Array of UINT	0-255	Set
7	Write the output data set 3 specific data	Array of UINT	0-255	Set
8	Write the output data set 4 specific data	Array of UINT	0-255	Set
9	Write the output data set 5 specific data	Array of UINT	0-255	Set

Common services

Tab. 43: Full Data Set Transfer object (72h) common services

Service code	Implemented in class	Implemented in instance	Service name
01h	Yes	Yes	Get_Attributes_All
0Eh	Yes	Yes	Get_Attribute_Single
10h	No	Yes	Set_Attribute_Single
02h	No	Yes	Set_Attribute_All

Individual Input Data Set Transfer object definition (73h — one instance per data set)

The vendor specific **Individual Input Data Set Transfer** object defines the attributes by which the PLC can request either full input data sets or individual parameters within an input data set.

Class attributes

Tab. 44: Individual Input Data Set Transfer object (73h) class attributes

Attribute ID	Name	Data type	Data values	Access rule
1	Revision	UINT	1	Get
2	Max instance	UINT	4	Get
3	Num instances	UINT	4	Get

Instance attributes

Tab. 45: Individual Input Data Set Transfer object (73h) instance attributes

Attribute ID	Name	Data type	Data values	Access rule
1 to *n* (dependent on data set definition)	Request input data set specific data	SINT	0-255	Get

Common services

Tab. 46: Individual Input Data Set Transfer object (73h) common services

Service code	Implemented in class	Implemented in instance	Service name
01h	Yes	Yes	Get_Attributes_All
0Eh	Yes	Yes	Get_Attribute_Single

Instance attribute definitions

Attribute 1 to n — Request input data set specific parameters
These attributes return the input data set specific data arrays. **Get Attribute Single** requests for a specific input data set will return only the requested data set parameter information. **Get Attribute All** requests will return the entire data set.

The data set attributes, numbered from 1 to N, refer to each individual attribute of each individual input data set. Each instance refers to a unique input data set and each input data set has a unique attribute numbering scheme. The following tables reflect the attribute definitions for each input data set.

Get All Data Set Attributes request

All data set information will be returned in integer (16 bit word) format. For byte data, the first byte will be placed in the least significant or rightmost byte of the integer and the second byte will be placed in the most significant or leftmost byte of the integer.

Example:

For an input data set, the data will be returned as follows:

- IntegerArray[0]: AABBh — AA = BYTE1; BB = BYTE2
- IntegerArray[1]: CCDDh — CC = MED1; DD = MED2
- ...
- IntegerArray[6]: MMNNh — MM = BYTE13; NN = BYTE14

Note The typical PC tools of Rockwell/Allen Bradley change this data format back to BBAA hex format for visualisation purposes. Check your data for plausibility before putting your Flexi Soft system into operation.

Instance 1 — Input data set 1 attribute definitions

Tab. 47: Individual Input Data Set Transfer object (73h) instance 1 attribute definitions

Attribute number	Data set parameter	Size
1	Byte 0	SINT
2	Byte 1	SINT
...	...	...
50	Byte 49	SINT

Instance 2 — Input data set 2 attribute definitions

Tab. 48: Individual Input Data Set Transfer object (73h) instance 2 attribute definitions

Attribute number	Data set parameter	Size
1	Overall CRC	UDINT
2	System CRC (SCID)	UDINT
3	Reserved	UDINT
4	Reserved	UDINT
5	Reserved	UDINT
6	Reserved	UDINT
7	Reserved	UDINT
8	Reserved	UDINT

Instance 3 — Input data set 3 attribute definitions

Tab. 49: Individual Input Data Set Transfer object (73h) instance 3 attribute definitions

Attribute number	Data set parameter	Size
1	Module status module 0	UINT[2]
2	Module status module 1	UINT[2]
...	...	...
15	Module status module 14	UINT[2]

Instance 4 — Input data set 4 attribute definitions

Tab. 50: Individual Input Data Set Transfer object (73h) instance 4 attribute definitions

Attribute number	Data set parameter	Size
1	Reserved	UINT[2]
2	Reserved	UINT[2]
...	...	...
15	Reserved	UINT[2]

PLC-5/SLC/MicroLogix interface

The PLC-5, SLC and MicroLogix PLC interfaces are supported by:

- the same write to PLC functionality as provided to ControlLogix PLCs provided in the Write-to-File receive method.
- PCCC based messages transferred via the PCCC object
  - SLC Typed Read Message
  - SLC Typed Write Message
  - PLC-5 Typed Read Message (Logical ASCII and Logical Binary address format)
  - PLC-5 Typed Write Message (Logical ASCII and Logic binary address format).
- Normal PLC-5/SLC file naming conventions are used.

The primary differences between the PLC-5/SLC/MicroLogix interface and the ControlLogix interfaces are:

- Polling is performed through the SLC and PLC-5 specific messages instead of accessing the Data Transfer object.
- Data is written into files on the PLC, instead of tags as on ControlLogix PLCs.

Note While ControlLogix PLCs support the SLC and PLC-5 messages, using those messages on ControlLogix PLCs is not recommended due to data size and performance considerations.

Receive communication methods

- Polling Receive Method

  This method provides a polling method that allows the PLC to request data on a periodic basis.

  In this method, the input data set information is returned in the response to the data request message. The PLC requests data by accessing the corresponding data file address on the FX0-GENT with either a SLC typed read or PLC-5 typed read message.

  The following restrictions apply to this method:

  - The file location to receive the input data set on the PLC must be of type INTEGER and large enough to contain the input data set table(s).
  - If no data has been received on the FLEXBUS+ for the specified module, all zeros will be returned.
- Unsolicited — Write to File Receive Method

  When it is determined that data received on the Flexi Soft gateway's FLEXBUS+ interface is to be sent to the PLC, the data is immediately written to a file location on the PLC.

  The following restrictions apply to this method:

  - The Receive Data Area File Name must have the same name as the file defined on the PLC. For SLC and PLC-5 PLCs, all file names must be configured with a preceding "$" (i.e $N10:0). For MicroLogix PLCs, all file names must be configured with a preceding "#" (i.e # N10:0).
  - The file on the PLC must be of type INTEGER and must be large enough to contain the input data set table(s).
  - Data will be written with the first byte placed in the MS byte location of the integer. **Example:** aabb, ccdd, eeff, etc. where aa = byte 1, bb = byte 2, cc = byte 3, etc.

Transmit (From PLC) Data Transfer Methods

The FX0-GENT will support the following methods of receiving or retrieving the output data set(s) from the PLC.

- PLC Writes Method

  This is the standard method where the PLC uses a message instruction to write the output data sets to the FX0-GENT. With this method, the output data sets can be updated via a PCCC message written to the corresponding file/address location on the FX0-GENT.

- Read-from-File Transmit Method (Poll the PLC)

  With this method, the FX0-GENT will monitor the configured PLC memory location for changes to the output data set(s). When a change is detected, the output data sets will be processed accordingly.

  The following restrictions apply to this method:

  – The output data set file locations must be of INTEGER (16 bit word) format and must be of sufficient length to contain the entire output data set.

  – Data in the INTEGER file must be formatted with the first byte placed in the MS byte location.
  Example: aabb, ccdd, eeff, etc. where aa = byte 1, bb = byte 2, cc = byte 3, etc.

PLC-5 and SLC Messages

The following PCCC messages are supported for the PLC-5, SLC and MicroLogix PLCs:

Tab. 51: Supported PCCC messages for the PLC-5, SLC and MicroLogix PLCs

Message type	PCCC message	Maximum message size
SLC Typed Read	162	CLX: 242 SINTs (121 INTs) SLC: 206 SINTs (103 INTs)
SLC Typed Write	170	CLX: 220 SINTs (110 INTs) SLC: 206 SINTs (103 INTs)
PLC-5 Typed Read	104	CLX: 234 SINTs (117 INTs) SLC: 252 SINTs (126 INTs)
PLC-5 Typed Write	103	CLX: 226 SINTs (113 INTs) SLC: 226 SINTs (113 INTs)

Note Both the PLC-5 and SLC Typed Read message can be used to retrieve all input data sets.

Tab. 52: Addressing for the PLC-5/SLC messages

Address	Description	Access rule	Data size [words]
N10:0	All enabled input data sets data	Get	16-101[7)]
N11:0	Request input data set 1 data	Get	25
N12:0	Request input data set 2 data	Get	16
N13:0	Request input data set 3 data	Get	30
N14:0	Request input data set 4 data	Get	30
N20:0	Write all enabled output data sets	Set	5-25[8)]
N21:0	Write output data set 1 data	Set	5
N22:0	Write output data set 2 data	Set	5
N23:0	Write output data set 3 data	Set	5
N24:0	Write output data set 4 data	Set	5
N25:0	Write output data set 5 data	Set	5

PLC-5/SLC Receive Data Message

The Receive Input Data Set Message format is as defined for each individual input data set. Please refer to section 3.3 “Data transmitted into the network (input data sets)” on page 14 for further details.

PCCC object (67h — 1 instance)

The **PCCC** object provides the ability to encapsulate and then transmit and receive PCCC messages between devices on an EtherNet/IP network. This object is used to communicate to SLC 5/05 and PLC-5 PLCs over EtherNet/IP.

Class attributes

Not supported.

Instance attributes

Not supported.

Instances

Supports instance 1.

Common services

Tab. 53: PCCC object (67h) common services

Service code	Implemented in class	Implemented in instance	Service name
4Bh	No	Yes	Execute_PCCC

7) Will correspond to all enabled input data sets.
8) Must correspond to all enabled output data sets. Example: If only output data sets 1 and 2 are enabled, then 10 words (20 bytes) must be written. If all output data sets are enabled, then 25 words (50 bytes) must be written.

Message structure for Execute_PCCC

Tab. 54: PCCC object (67h) request message

Name	Data type	Description
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial number	UDINT	ASA serial number of requestor
CMD	USINT	Command byte
STS	USINT	0
TNSW	UINT	Transport word
FNC	USINT	Function code
PCCC_params	Array of USINT	CMD/FMC specific parameters

Tab. 55: PCCC object (67h) response message

Name	Data type	Description
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial number	UDINT	ASA serial number of requestor
CMD	USINT	Command byte
STS	USINT	Status byte
TNSW	UINT	Transport word. Same value as request
EXT_STS	USINT	Extended status (if error)
PCCC_params	Array of USINT	CMD/FMC specific result data

Tab. 56: PCCC object (67h) supported PCCC command types

CMD	FNC	Description
0Fh	67h	PLC-5 write
0Fh	68h	PLC-5 read
0Fh	A2h	SLC 500 protected read with 3 address fields
0Fh	AAh	SLC 500 protected write with 3 address fields

5.2.7 Example for the configuration of explicit messaging

This section gives an example how to configure explicit messaging using RSLogix.

Required gateway settings

In the **Gateway configuration** dialog of the Flexi Soft Designer, the following settings have to be activated:

- PLC requests
- PLC writes

Required RSLogix settings

In RSLogix, the following settings must be made:

- PLC is Active (explicit messaging enabled)
- 128 connections possible
- Each Data Set must have the correct size.
- Main program sends a message with command to SET or GET (either Get_Attribute_Single or Get_Attributes_All, see Tab. 46).

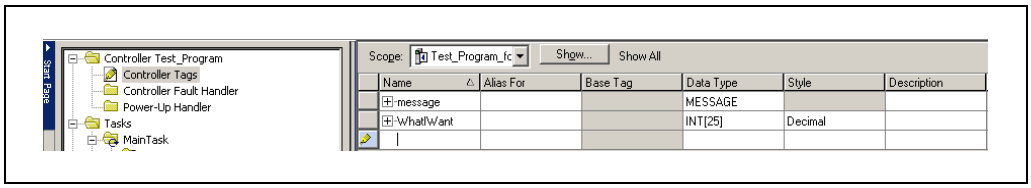
Step 1:

➢ Create two tags. One for the MESSAGE and one to store the data from the GET command.

The MESSAGE tag will be for the MSG block which is used for explicit messaging. The MSG command in this example will request Data Set 1. The received Data Set 1 will then be placed into a tag of our choosing, called “WhatIWant”.

Note The destination tag must be set to the size of the Data Set. In this case, it would be 50 bytes or INT[25].

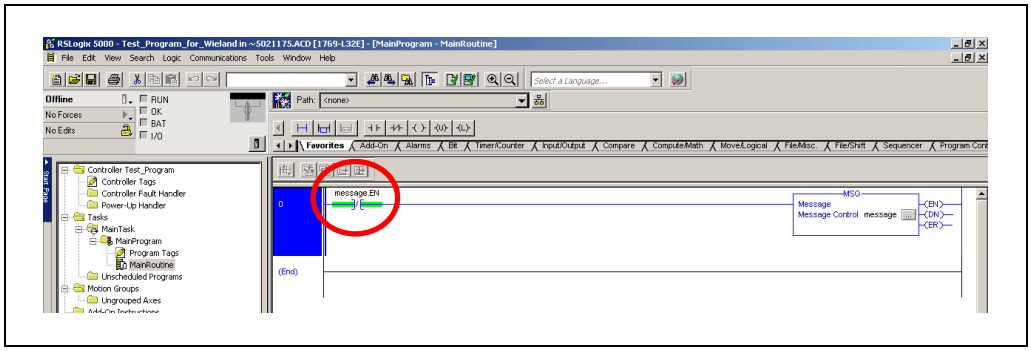
Fig. 31: Creating tags for explicit messaging



Step 2:

➢ On one line in the program, create a NOT connected to the MSG command

Fig. 32: Main routine programming for explicit messaging



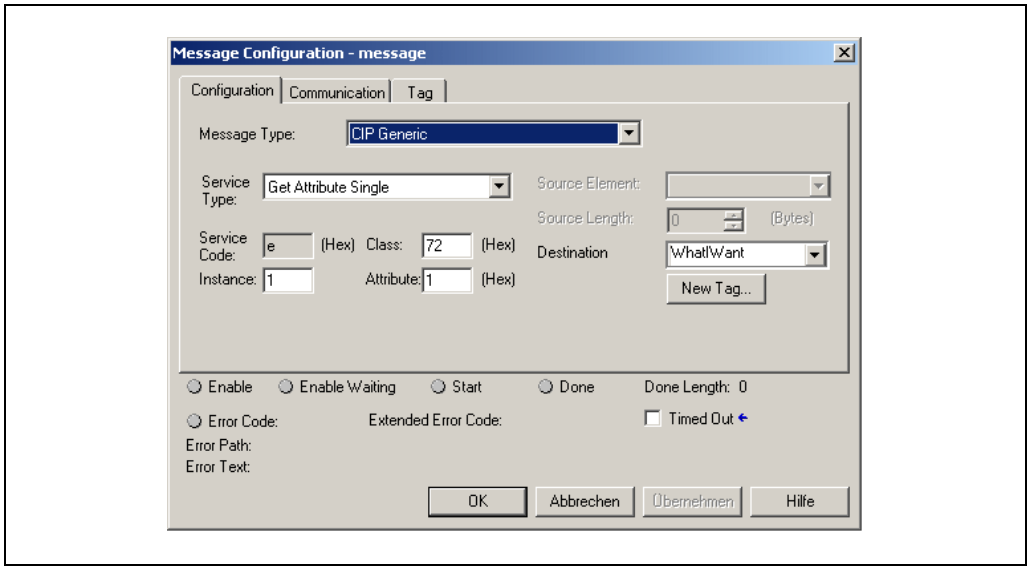
➢ Select “message.EN” for the NOT symbol. This ensures that the message will be requested repeatedly.

Step 3:

➢ Edit the message command as follows:

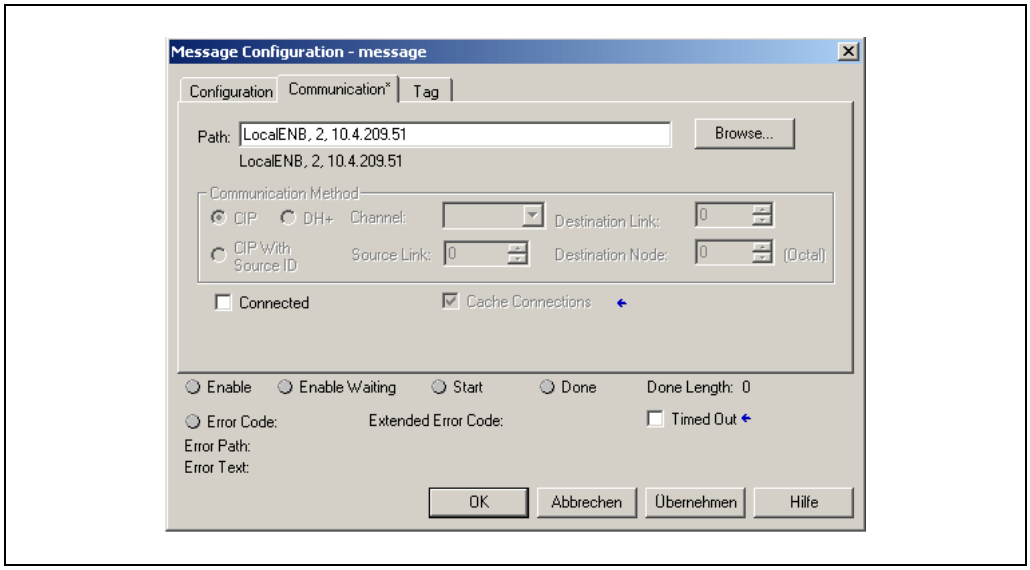
– Configuration: Select **CIP Generic**. The **Class** describes the object. **Class 72** is for *Full Data Set Transfer*. **Instance 1** and **Attribute 1** determine the data type which in this case is **Data Set 1** (see Tab. 42). As **Destination** the “WhatIWant” tag must be chosen.

Fig. 33: Explicit messaging — Message configuration



– Communication: The message has to contain the **Path** to the gateway. In this example the path is 10.4.209.51.

Fig. 34: Explicit messaging — Communication configuration



5.2.8 TCP/IP configuration interface

See section 5.1.1 “TCP/IP configuration interface” on page 32.

5.2.9 Ethernet TCP/IP socket interface

See section 5.1.2 “Ethernet TCP/IP socket interface” on page 37.

5.2.10 Diagnostics and troubleshooting

For information how to perform diagnostics on the Flexi Soft system please refer to the operating instructions for the Flexi Soft Designer software (SICK part no. 8012998).

Tab. 57: Troubleshooting for the FX0-GENT

Symbol description:
○: LED is off
● Green: LED lights up green
☼● Red: LED flashes red

Error	Possible cause	Possible remedy
The Flexi Soft Designer tool does not connect to the Flexi Soft gateway module	FX0-GENT has no power supply. FX0-GENT is not in the same physical network as the PC. The PC is configured to another subnet mask in the TCP/IP settings. FX0-GENT has already been configured once and has a fixed set IP address or an IP address assigned by a DHCP server that is not recognised.	Establish the power supply. Check the Ethernet wiring and network settings on the PC and correct if necessary. Set the subnet mask on the PC to 255.255.0.0 (factory setting of the FX0-GENT). Check the communication settings in the Flexi Soft Designer.
FX0-GENT does not supply any data. PWR ● Green LINK/ACT ●/☼● Green STATUS[9] ☼● Red/green	FX0-GENT is configured for data transfer to PLC, but Ethernet communication is not yet established or faulty. Duplicate IP address detected. Another device on the network has the same IP address.	Minimum one Ethernet connection needs to be established. Set up Ethernet connection on PLC side, check Ethernet cabling, check Ethernet connection settings on PLC and in the Flexi Soft Designer. If no Ethernet communication is required, disable the Ethernet connections/PLC interfaces on the FX0-GENT. Adjust IP address and power cycle device.
FX0-GENT does not supply any data. PWR ● Green LINK/ACT ●/☼● Green STATUS[9] ☼● Red (1 Hz)	Configuration required. Configuration download is not completed.	Configure the FX0-GENT and download the configuration to the device. Wait until the configuration download has been completed.
FX0-GENT does not supply any data. PWR ● Green LINK/ACT ●/☼● Green STATUS[9] ● Green	No data set is activated. No Ethernet communication interface is enabled.	Activate at least one data set.
FX0-GENT does not supply any data. PWR ● Green LINK/ACT ●/☼● Green STATUS[9] ☼● Green (1 Hz)	FX0-GENT is in Idle mode.	CPU/application is stopped. Start CPU (change into Run state).
FX0-GENT functioned correctly after configuration, but suddenly no longer supplies data. PWR ● Green LINK/ACT ●/☼● Green STATUS[9] ☼● Red/green	FX0-GENT is operated in slave mode, the IP address is assigned from a DHCP server. After the FX0-GENT or the DHCP server has been restarted, a different IP address that is unknown to the PLC has been assigned to the FX0-GENT.	Either assign a fixed IP address to the FX0-GENT, or reserve a fixed IP address for the FX0-GENT in the DHCP server (manual assignment by means of the MAC address of the FX0-GENT).
FX0-GENT/Flexi Soft system is in Critical fault mode. PWR ● Green LINK/ACT ☼● Green STATUS[9] ● Red	FX0-GENT is not plugged properly into the other Flexi Soft module. Module connecting plug is soiled or damaged. Other Flexi Soft module has internal critical error.	Plug the FX0-GENT in correctly. Clean the connecting socket/plug. Repower the system. Check the other Flexi Soft modules.
FX0-GENT is in Critical fault mode. PWR ● Green LINK/ACT ●/☼● Green STATUS[9] ☼● Red (2 Hz)	FX0-GENT internal device error CPU firmware version does not support Flexi Soft gateways.	Switch off the power supply of the Flexi Soft system and switch it on again. Check the diagnostics messages with the Flexi Soft Designer. Use a FX3-CPUx with the required firmware version (see section 2.2 “Correct use” on page 10). If the error remains, replace the gateway.

[9] On older versions of the FX0-GENT, the STATUS LED is called MS LED.

5.3 Modbus TCP gateway

The following Flexi Soft gateway can be used for Modbus/TCP: FX0-GMOD.

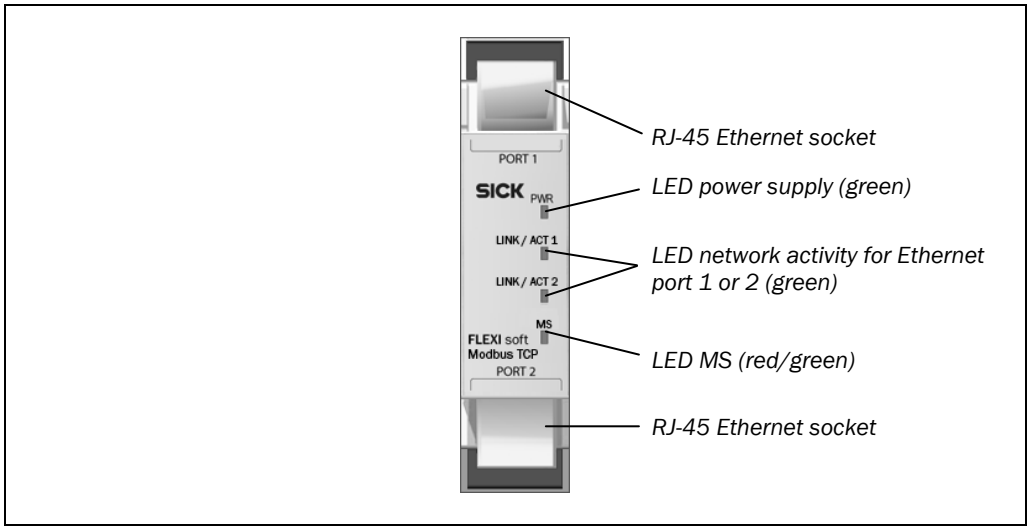
The Flexi Soft Modbus TCP gateway supports:

- Modbus TCP master and slave receive methods
- Ethernet TCP/IP socket interface, polling and auto update function
- With firmware ≥ V2.01.0: Data can be read also wordwise.

5.3.1 Interfaces and operation

The FX0-GMOD is equipped with an integrated three-port switch for connection with the Ethernet network. Two RJ-45 sockets are available for the connection. The switch functionality allows the FX0-GMOD to be used for connection to another Ethernet component (e.g. connection to a notebook) without having to interrupt the Ethernet connection to the network.

Fig. 35: Interfaces and display elements of the FX0-GMOD



Tab. 58: Meaning of the LED displays

Symbol description:
○: LED is off
● Green: LED lights up green
☼● Red: LED flashes red

LED		Meaning
PWR	○	No power supply
	● Green	Power supply switched on
LINK/ACT 1 LINK/ACT 2	○	No Ethernet connection
	● Green	Ethernet connection active, no data transmission
	☼● Green	Ethernet connection active, data transmission
MS	○	Power-up
	● Green	Executing (live process data from/to FX3-CPUx)
	☼● Green	1 Hz: Idle
	☼● Red	1 Hz: Configuring/configuration required 2 Hz: Critical fault on gateway
	● Red	Critical fault on another module
	☼● Red/green	Executing, but Ethernet communication not established or faulty

Note Error elimination is described in section 5.3.6 “Diagnostics and troubleshooting” on page 83.

Power-up sequence

On power-up, the following LED test sequence is performed:

- MS LED ○ Off for 6 s
- MS LED ● Red for 0.25 s
- MS LED ● Green for 0.25 s
- MS LED ○ Off

5.3.2 Basic configuration — Assigning an IP address

Configuration of the FX0-GMOD is performed via the Flexi Soft Designer tool.

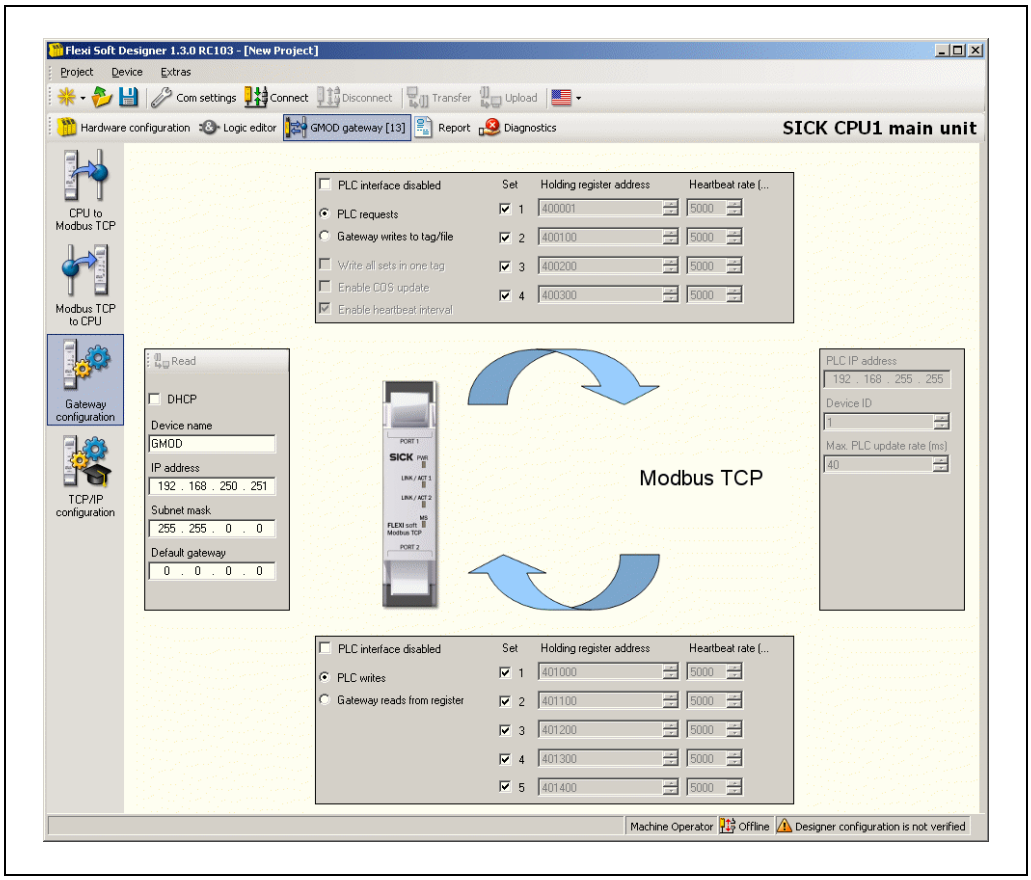
Via Flexi Soft Designer tool

➢ Open the Flexi Soft Designer and load the hardware configuration including the Modbus TCP gateway.

➢ Click on the **Gateways** button above the main window and select the FX0-GMOD or double click the FX0-GMOD in the hardware configuration view to open the gateway configuration dialog.

➢ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 36: Modbus TCP gateway configuration dialog



➢ Enter a valid **IP address**, **Subnet mask** and if required a valid IP address for a **Default gateway**.

➢ Click **OK**.

➢ Click **Connect** to go online and download the configuration to the Flexi Soft system.

5.3.3 Configuration of the Modbus TCP interface to the PLC — How the data are transferred

Modbus TCP application characteristics:

- support of standard Modbus TCP addressing conventions
- Master and Slave receive methods

Modbus TCP PLC requirements:

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

The configuration steps in this section specify how the data to the higher-level PLC are transferred.

In general, there are two different transfer methods available for each transfer direction such as *Flexi Soft to Network* and *Network to Flexi Soft*:

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

  This method provides a polling method that allows the PLC to request data on a periodic basis. In this method, the data is returned in the response to the data request message. The PLC requests data by accessing the receive data address on the FX0-GMOD module with a Read Holding Registers message.
- Master receive method — Gateway writes to PLC (auto-update, gateway as master)

  When it is determined that data received on the FX0-GMOD module backplane interface is to be sent to the PLC, the data is immediately written to a data memory location on the PLC.
- Slave transmit method — PLC writes (gateway as slave)

  In this method, the PLC will send write messages to the FX0-GMOD module to set the output data sets. To write to the output data sets, the PLC writes the data to specified addresses.
- Master transmit method — Gateway reads from PLC (auto-update, gateway as master)

  In the master transmit mode, the FX0-GMOD module will poll the PLC for the output data set settings.

Note The configuration is considered faulty, if the PLC IP address is zero and either the Read Transfer mode and/or the Write Transfer mode is set to Master.

The number of possible connections to the PLC depends on whether the FX0-GMOD is operated as a master or slave. Depending on the setting, up to 32 PLCs can address the FX0-GMOD at the same time.

Tab. 59: Number of possible connections

Operating mode of the FX0-GMOD	Maximum connections
Rx (To PLC) transfer mode: Master Tx (From PLC) transfer mode: Master	Rx and Tx: 1
Rx (To PLC) transfer mode: Master Tx (From PLC) transfer mode: Slave	Rx: 1 Tx: 31
Rx (To PLC) transfer mode: Slave Tx (From PLC) transfer mode: Master	Rx: 31 Tx: 1
Rx (To PLC) transfer mode: Slave Tx (From PLC) transfer mode: Slave	Rx and Tx: 32

The following table outlines a guideline to the configuration process depending on the transfer method:

Tab. 60: Configuration guideline — Gateway as master

Gateway is master	
To do in the gateway configuration (via Flexi Soft Designer tool)	**To do in the PLC program and/or Modbus TCP network configuration tool**
Select **Gateway writes to tag/file** and/or **Gateway reads from register** to configure gateway as master.	–
Select which data shall be written to/read from the PLC	–
Define where in the PLC memory the selected data shall be written to: Enter holding register address(es). Example: “400001” And/or define where in the PLC memory the selected data shall be read from: Enter holding register addresses.	Ensure the addresses defined in the Flexi Soft Designer configuration are available and contain the data determined for the Flexi Soft system. **Note:** Modbus TCP communication uses port 502 as standard port.
Select how often this data shall be transmitted.	–
Define where the data shall be read from/written to in the Modbus TCP network: Enter the IP address and controller slot number of the PLC.	–

Tab. 61: Configuration guideline — Gateway as slave

Gateway is slave	
To do in the gateway configuration (via Flexi Soft Designer tool)	**To do in the PLC program and/or Modbus TCP network configuration tool**
Select **PLC requests** and **PLC writes** in the gateway configuration dialog	–
–	Define which data shall be written to/read from the gateway. Ensure the PLC program writes the data into the addresses defined for the gateway (see Tab. 62 on page 82).

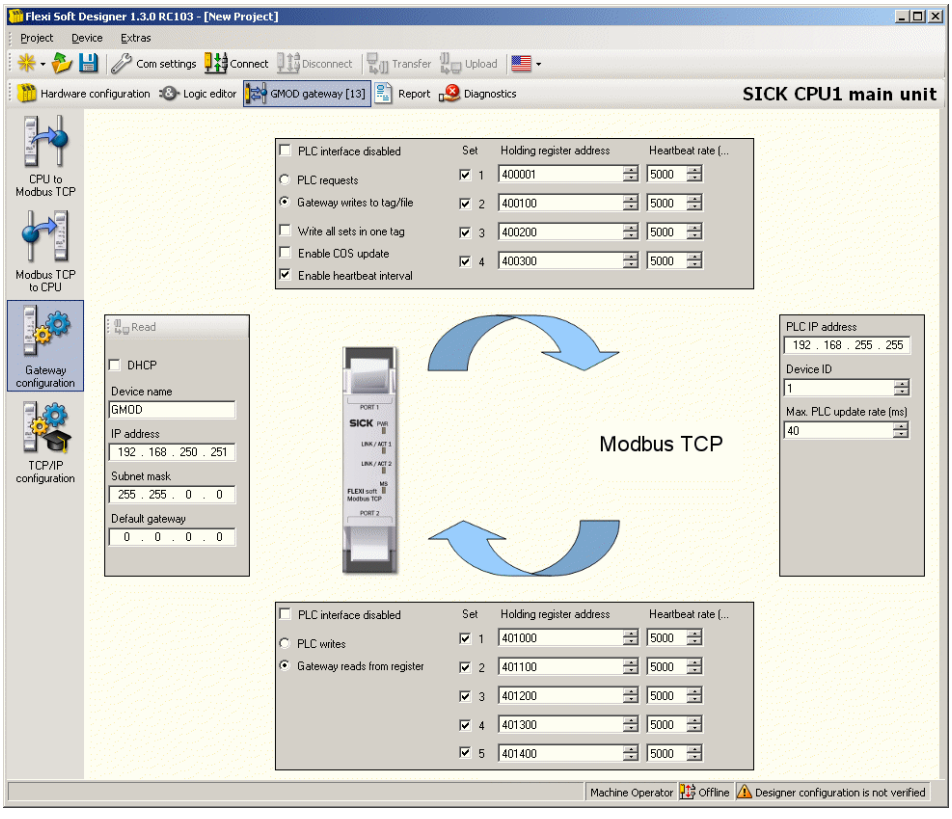
Note The Modbus TCP gateway address setting is based 1. Please add 1 to the holding register address set in the Flexi Soft Designer for an address setting based 0.

Master mode — FX0-GMOD reads from/writes to the PLC

In order to configure the gateway to be *master*, perform the following steps:

- ➢ Open the Flexi Soft Designer and load the hardware configuration including the Modbus TCP gateway.
- ➢ Click on the **Gateways** button above the main window and select the FX0-GMOD or double click the FX0-GMOD in the hardware configuration view to open the gateway configuration dialog.
- ➢ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 37: Modbus TCP gateway configuration as master



- ➢ Within the **Gateway configuration** dialog, select the transfer method by checking **Gateway writes to tag/file** for the Flexi Soft to Network direction and **Gateway reads from register** for the Network to Flexi Soft direction.
- ➢ Select which data shall be written to/read from the PLC by checking the checkbox for the required data set. For the exact description of the data sets please refer to section 3.3 “Data transmitted into the network (input data sets)” on page 14.

- Define where in the PLC memory the selected data shall be written to or read from: Enter addresses into the **Holding register address** field (max. 20 characters).
- Select **Write all sets in one tag** if all data sets shall be written into one address in the PLC memory. In this case, the tag/file defined for data set 1 will be used.
- For the **Flexi Soft to network** direction, define how often the data shall be transmitted:
  - Select **Enable COS update** if the FX0-GMOD is to update the data in the PLC immediately when changes occur in the data sets.
  - Select **Enable heartbeat interval** to activate updating of the selected data sets with the set Heartbeat rate.

  Both options may be selected at the same time.
- For the **Network to Flexi Soft** direction, define how often the data shall be read:
  - Enter a **Heartbeat rate** to activate updating of the selected data sets with the set time interval.
- Define where the data shall be read from/written to in the Modbus TCP network: Enter the **PLC IP address** and the Modbus **Device ID** of the PLC.
- **Max. PLC update rate** defines the maximum rate (the minimum time interval) for sending the data sets to the PLC. Settings occur dependent on the PLC processing speed. Minimum = 10 ms, maximum = 65,535 ms. The default value of 40 ms is suitable for most PLCs.

Note If this value is greater than the **Heartbeat rate**, the heartbeat rate is slowed down to this value.

- Go online and download the configuration to the Flexi Soft system.

Write to PLC

Note The following restrictions apply when the gateway is master and writes the input data sets to the PLC:

- The input data set address (set via Flexi Soft Designer Tool) must be the same as that defined on the PLC.
- The variable to receive the data on the PLC must be:
  - in the 40xxxx address range (for Schneider Modicon type PLCs)
  - an array of 16 bit words
  - long enough to contain the specified input data set array.
- All input data sets are transferred to the PLC in 16 bit word format with the first byte placed in the least significant, or rightmost byte of the integer and the second byte placed in the most significant, or leftmost byte of the integer.

Read from PLC

Note The following restrictions apply when the gateway is master and reads the output data sets from the PLC:

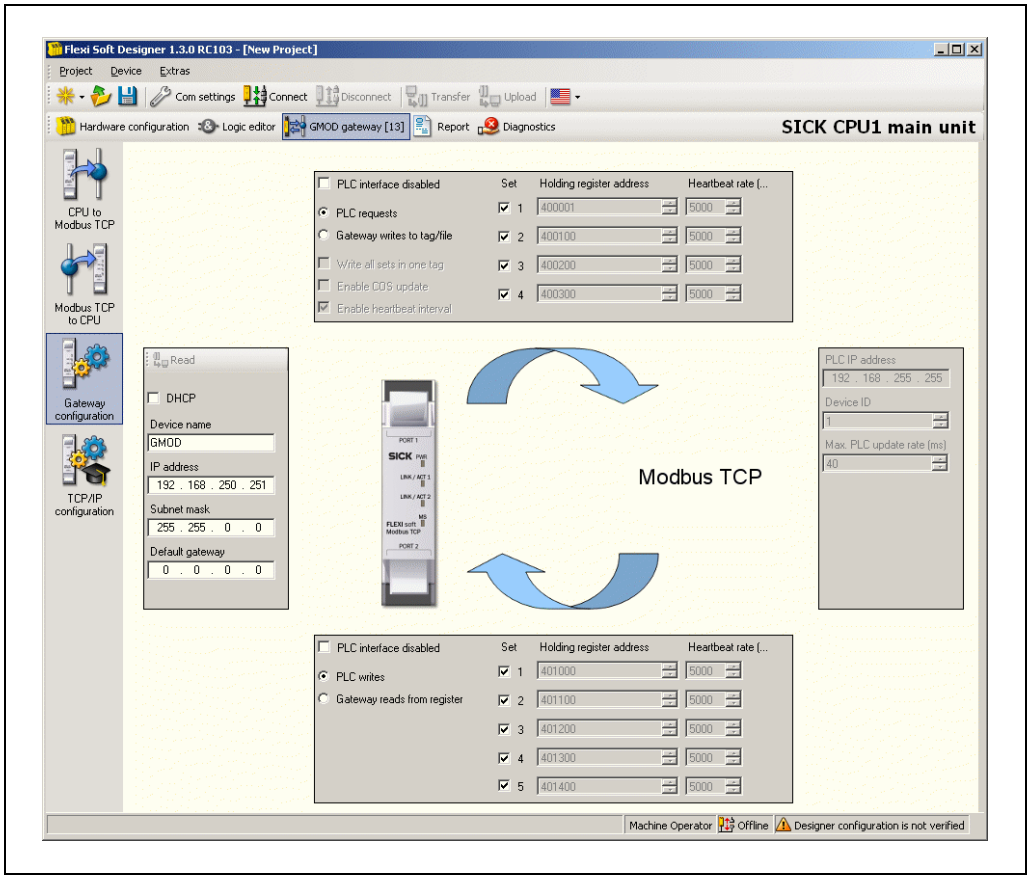
- The output data set addresses must be the same as those defined on the PLC.
- The variables to request the data on the PLC must be:
  - in the 40xxxx address range (for Schneider Modicon type PLCs)
  - for the output data set settings, an array of 16 bit words long enough to contain the entire output data set.
- All output data sets are transferred from the PLC in 16 bit word format and the first byte must be placed in the least significant, or rightmost byte of the integer and the second byte placed in the most significant, or leftmost byte of the integer.

Slave mode — PLC reads from/writes to the FX0-GMOD

In this operating mode the FX0-GMOD sends the data as *slave* upon request from the PLC. If this operating mode is desired:

- Open the Flexi Soft Designer and load the hardware configuration including the Modbus TCP gateway.
- Click on the **Gateways** button above the main window and select the FX0-GMOD or double click the FX0-GMOD in the hardware configuration view to open the gateway configuration dialog.
- Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 38: Modbus TCP gateway configuration as slave



- Within the **Gateway configuration** dialog, select the transfer method by checking **PLC requests** for the Flexi Soft to Network direction and **PLC writes** for the Network to Flexi Soft direction.
- Select which data shall be written/read to/from the PLC by checking the checkbox for the required data set. For the exact description of the data sets please refer to section 3.3 “Data transmitted into the network (input data sets)” on page 14.
- Click **OK**.
- Go online and download the configuration to the Flexi Soft system.

PLC writes output data sets

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

- The device index must be 1.
- The message must be sent in word format.
- All output data sets are transferred from the PLC in 16 bit word format and the first byte must be placed in the least significant, or rightmost byte of the integer and the second byte placed in the most significant, or leftmost byte of the integer.

PLC polls input data sets

The following restrictions apply to this method:

- The device index must be 1.
- The variable to receive the data on the PLC must be:
  - in the 40xxxx address range (for Schneider Modicon type PLCs)
  - an array of 16 bit words
  - long enough to contain the data set array(s)
- All input data sets are transferred to the PLC in 16 bit word format with the first byte placed in the least significant, or rightmost byte of the integer and the second byte placed in the most significant, or leftmost byte of the integer.

FX0-GMOD as slave — Data addressing

The following table lists the address to read out the data sets.

Tab. 62: Data addressing for FX0-GMOD as receiver

Unit ID	1

Address (Base 1)	Description	Access	Scope [words]
1000	Request all enabled input data sets	Get	16-101[10)]
1100	Request input data set 1 data	Get	25
1200	Request input data set 2 data	Get	16
1300	Request input data set 3 data	Get	30
1400	Request input data set 4 data	Get	30
2000	Write all enabled output data sets data	Set	5-25[11)]
2100	Write output data set 1 data	Set	5
2200	Write output data set 2 data	Set	5
2300	Write output data set 3 data	Set	5
2400	Write output data set 4 data	Set	5
2500	Write output data set 5 data	Set	5

Note All data sets can only be read or written as a block. It is not possible to read or write single bits or bytes.

[10)] Will correspond to all enabled input data sets.
[11)] Must correspond to all enabled output data sets. Example: If only output data sets 1 and 2 are enabled, then 10 words (20 bytes) must be written. If all output data sets are enabled, then 25 words (50 bytes) must be written.

Modbus commands and error messages

The FX0-GMOD supports the following Modbus commands and error messages:

Tab. 63: Modbus commands

Modbus command	Value
Read holding registers	3
Write multiple registers	16 (10h)
Read/write multiple registers	23 (17h)

Tab. 64: Modbus error messages

Modbus error response	Description
1 Illegal function	The requested function is not supported
2 Illegal data address	Undefined data address received
3 Illegal data value	Request with illegal data values, for example not enough data requested for a data set
10 Gateway paths not available	Invalid configuration, for example polling or setting of the digital outputs via PLC during operation of the FX0-GMOD in master mode

5.3.4 TCP/IP configuration interface

See section 5.1.1 “TCP/IP configuration interface” on page 32.

5.3.5 Ethernet TCP/IP socket interface

See section 5.1.2 “Ethernet TCP/IP socket interface” on page 37.

5.3.6 Diagnostics and troubleshooting

For information how to perform diagnostics on the Flexi Soft system please refer to the operating instructions for the Flexi Soft Designer software (SICK part no. 8012998).

Tab. 65: Troubleshooting for the FX0-GMOD

Symbol description:
○: LED is off
● Green: LED lights up green
☼ Red: LED flashes red

Error	Possible cause	Possible remedy
The Flexi Soft Designer tool does not connect to the Flexi Soft gateway module	FX0-GMOD has no power supply. FX0-GMOD is not in the same physical network as the PC. The PC is configured to another subnet mask in the TCP/IP settings. FX0-GMOD has already been configured once and has a fixed set IP address or an IP address assigned by a DHCP server that is not recognised.	Establish the power supply. Check the Ethernet wiring and network settings on the PC and correct if necessary. Set the subnet mask on the PC to 255.255.0.0 (factory setting of the FX0-GMOD). Check the communication settings in the Flexi Soft Designer.
FX0-GMOD does not supply any data. PWR ● Green LINK/ACT ●/☼ Green MS ☼ Red/green	FX0-GMOD is configured for data transfer to PLC, but Ethernet communication is not yet established or faulty. Duplicate IP address detected. Another device on the network has the same IP address.	Minimum one Ethernet connection needs to be established. Set up Ethernet connection on PLC side, check Ethernet cabling, check Ethernet connection settings on PLC and in the Flexi Soft Designer. If no Ethernet communication is required, disable the Ethernet connections/PLC interfaces on the FX0-GMOD. Adjust IP address and power cycle device. For Modbus TCP communication check the Modbus port number at the PLC (the Modbus port number must be 502). Do not confuse the Modbus port number with the TCP/IP socket port number which must be > 1023.

Error	Possible cause	Possible remedy
FX0-GMOD does not supply any data. PWR ● Green LINK/ACT ●/☼ Green MS ☼ Red (1 Hz)	Configuration required. Configuration download is not completed.	Configure the FX0-GMOD and download the configuration to the device. Wait until the configuration download has been completed.
FX0-GMOD does not supply any data. PWR ● Green LINK/ACT ●/☼ Green MS ● Green	No data set is activated. No Ethernet communication interface is enabled.	Activate at least one data set.
FX0-GMOD does not supply any data. PWR ● Green LINK/ACT ●/☼ Green MS ☼ Green (1 Hz)	FX0-GMOD is in Idle mode.	CPU/application is stopped. Start CPU (change into Run state).
FX0-GMOD functioned correctly after configuration, but suddenly no longer supplies data. PWR ● Green LINK/ACT ●/☼ Green MS ☼ Red/green	FX0-GMOD is operated in slave mode, the IP address is assigned from a DHCP server. After the FX0-GMOD or the DHCP server has been restarted, a different IP address that is unknown to the PLC has been assigned to the FX0-GMOD.	Either assign a fixed IP address to the FX0-GMOD, or reserve a fixed IP address for the FX0-GMOD in the DHCP server (manual assignment by means of the MAC address of the FX0-GMOD).
FX0-GMOD/Flexi Soft system is in Critical fault mode. PWR ● Green LINK/ACT ☼ Green MS ● Red	FX0-GMOD is not plugged properly into the other Flexi Soft module. Module connecting plug is soiled or damaged. Other Flexi Soft module has internal critical error.	Plug the FX0-GMOD in correctly. Clean the connecting socket/plug. Repower the system. Check the other Flexi Soft modules.
FX0-GMOD is in Critical fault mode. PWR ● Green LINK/ACT ●/☼ Green MS ☼ Red (2 Hz)	FX0-GMOD internal device error CPU firmware version does not support Flexi Soft gateways.	Switch off the power supply of the Flexi Soft system and switch it on again. Check the diagnostics messages with the Flexi Soft Designer. Use a FX3-CPUx with the required firmware version (see section 2.2 “Correct use” on page 10). If the error remains, replace the gateway.

5.4 PROFINET IO gateway

The following Flexi Soft gateway can be used for PROFINET IO: FX0-GPNT.

You will find the GSDML file and device icon for PLC interfacing with PROFIBUS support …

- in the Internet on the FX0-GPNT product page on www.sick.com.
- in the Flexi Soft Designer program folder on your hard disk (default installation folder is “C:\programs\SICK\FlexiSoft\DeviceDescriptions\…”).

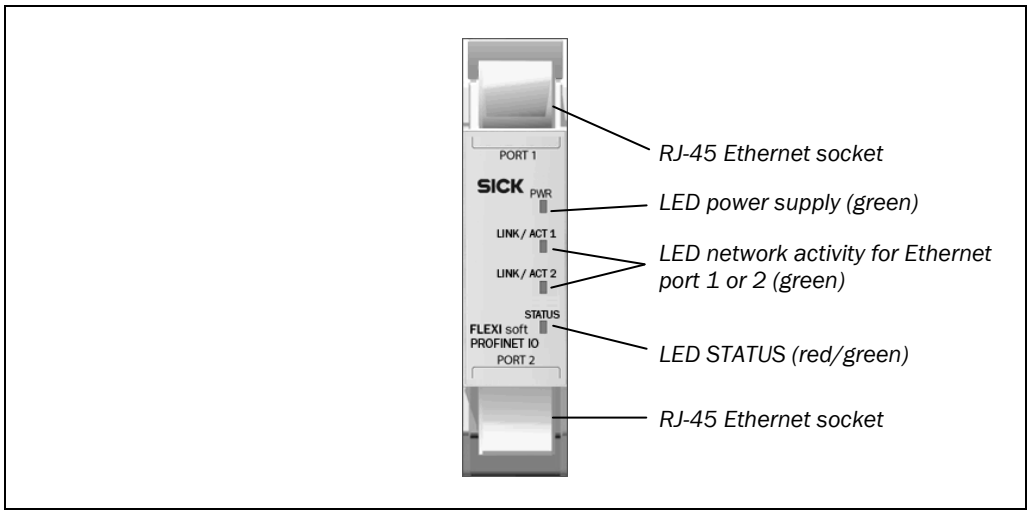
The FX0-GPNT supports

- PROFINET IO conformance class A
- LLDP
- SNMP
- MIB II
- Fast integrated switching
- Auto-MDI
- Auto negotiation
- Cyclic I/O communication

5.4.1 Interfaces and operation

The FX0-GPNT is equipped with an integrated 3-port switch for connection with the Ethernet network. Two RJ-45 sockets are available for the connection. The switch functionality allows the FX0-GPNT to be used for connection to another Ethernet component (e.g. connection to a notebook) without having to interrupt the Ethernet connection to the network.

Fig. 39: Interfaces and display elements of the FX0-GPNT



Tab. 66: Meaning of the LED displays of the FX0-GPNT

Symbol description:
○: LED is off
● Green: LED lights up green
☼● Red: LED flashes red

LED		Meaning
PWR	○	No power supply
	● Green	Power supply switched on
LINK/ACT 1 LINK/ACT 2	○	No Ethernet connection
	● Green	Ethernet connection active, no data transmission
	☼● Green	Ethernet connection active, data transmission
STATUS	○	Power-up
	● Green	Executing (live process data from/to FX3-CPUx)
	☼● Green	1 Hz: Idle 2 Hz: PROFINET master requested LED flashing for physical device identification
	☼● Red	1 Hz: Configuring/configuration required 2 Hz: Critical fault on gateway
	● Red	Critical fault on another module
	☼● Red/green	Executing, but Ethernet communication not established or faulty (can be switched off, see below)

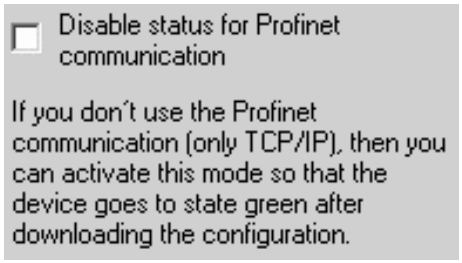
Note Error elimination is described in section 5.4.7 “Diagnostics and troubleshooting” on page 100.

Disabling the STATUS LED for PROFINET communication

With firmware ≥ V2.00.0 the red/green flashing of the STATUS LED can be disabled in the Flexi Soft Designer. Otherwise the LED will flash permanently if no PROFINET communication is established (e.g. if the gateway is used for TCP/IP communication only).

- Click on the **Gateways** button above the main window and select the FX0-GPNT or double click the FX0-GPNT in the hardware configuration view to open the gateway configuration dialog.
- Click on **Gateway configuration** on the left hand menu. On the right side of the window you will find the following configuration panel:

Fig. 40: Disabling the STATUS LED of the FX0-GPNT



- Activate the checkbox. After downloading the configuration, the LED will light up ● Green permanently, even if no PROFINET communication is established.

Note This feature is only available with firmware ≥ V2.00.0 and Flexi Soft Designer version ≥ 1.4.0.

Power-up sequence

On power-up, the following LED test sequence is performed:

- STATUS LED ○ Off for 6 s
- STATUS LED ● Red for 0.25 s
- STATUS LED ● Green for 0.25 s
- STATUS LED ○ Off

5.4.2 Basic configuration — Assigning a device name and IP address

Configuration and diagnostic of the FX0-GPNT may be performed via the Flexi Soft Designer tool as well as with the PROFINET IO network programming tool (e.g. SIEMENS SIMATIC).

Configuration via PROFINET IO

In the out-of-the-box configuration, each PROFINET IO field device, e.g. the FX0-GPNT has a MAC address and a symbolic name stored.

Notes

- The symbolic name for the gateway is **FX0-GPNT**.
- This name is used by the I/O controller (i.e. PLC) to assign the IP address to the field device.
- If the IP address is also used for other Ethernet communications like TCP/IP or configuration over Ethernet, remember that the PLC can change the IP address so these can be interupted.

Address assignment is performed in two steps.

- Assign a unique plant specific name to the gateway by using either the network configuration tool, e.g. SIEMENS SIMATIC Manager or the Flexi Soft Designer tool.
- Using the plant specific (unique) name, the I/O controller (i.e. PLC) can now assign the IP address to the gateway before system boot.

Note The FX0-GPNT MAC address is printed on the device type label (example: 00:06:77:02:00:A7).

Device name set via SIEMENS SIMATIC Manager

Refer to section “Step 4: Assign the device name” on page 91.

Device name set via Flexi Soft Designer

- Open the Flexi Soft Designer and load the hardware configuration including the PROFINET IO gateway. Ensure your project is offline.
- Click on the **Gateways** button above the main window and select the FX0-GPNT or double click the FX0-GPNT in the hardware configuration view to open the gateway configuration dialog.
- Click on **Gateway configuration** on the left hand menu. The following dialog appears:

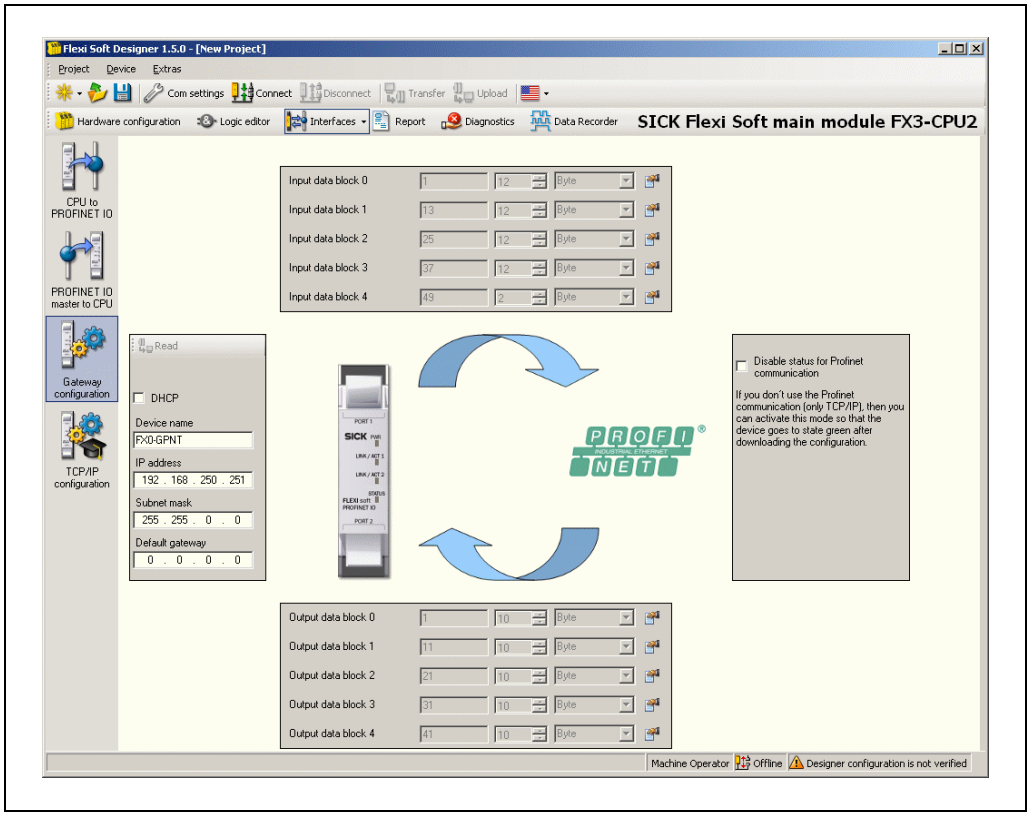


Fig. 41: PROFINET IO gateway configuration dialog

- Enter the device name in the **Device name** field (maximum length 255 characters).

Notes

- The device name format must apply to the PROFINET standard specification.
- Ensure that the default gateway address corresponds to the address set by the PLC for the gateway. If there is no router used, Siemens Step 7 uses as default gateway address the same address as the IP address for the GPNT.

IP address set via Flexi Soft Designer tool

Usually the IP address will be assigned by the PROFINET IO controller (e.g. PLC). However, the FX0-GPNT allows configuration of the entire Flexi Soft system over Ethernet TCP/IP. In this case, it may be necessary to assign an IP address to the gateway even before the PROFINET IO network has been setup. This can be done on the configuration page shown in Fig. 41 as well.

5.4.3 PROFINET configuration of the gateway — How the data are transferred

The following steps need to be taken in order to configure the communication between PLC and gateway.

Note This document does not cover the creation of the PROFINET IO network or the rest of the automation system project in the network configuration tool. It is assumed the PROFINET project has already been set up in the configuration program, e.g. SIEMENS SIMATIC Manager. Examples refer to configurations performed with SIEMENS SIMATIC manager.

Step 1: Install the generic station description file (GSDML file)

Before the FX0-GPNT can be used as device in the network configuration tool, e.g. SIEMENS SIMATIC Manager, for the first time, the generic station description (GSDML) of the gateway must be installed into the hardware catalogue of the tool.

- Download the GSDML file and device icon from www.sick.com, on the FX0-GPNT product page.
- Follow the instructions in the online help or in the user manual of the PROFINET network configuration tool for installing generic station description files.

Using the SIEMENS SIMATIC Manager — HW Config, the gateway then appears in the hardware catalogue under >>**PROFINET IO** > **Additional Field Devices** > **Gateway** > **FlexiSoft**.

Step 2: Add the gateway to the project

In order to have the Flexi Soft system data available in the PLC process image, the gateway must be added to the hardware configuration first. The procedure associated with this depends on the hardware configuration program of the PLC being used. On this topic, please also read the documentation for the corresponding program.

The example below shows how to add the gateway to a SIEMENS SIMATIC Manager project.

In the SIEMENS SIMATIC Hardware Manager, the gateway can be found in the hardware catalogue under >>**PROFINET IO** > **Additional Field Devices** > **Gateway** > **FlexiSoft**.

➢ Drag & drop the device into the Ethernet PROFINET IO network. Example:

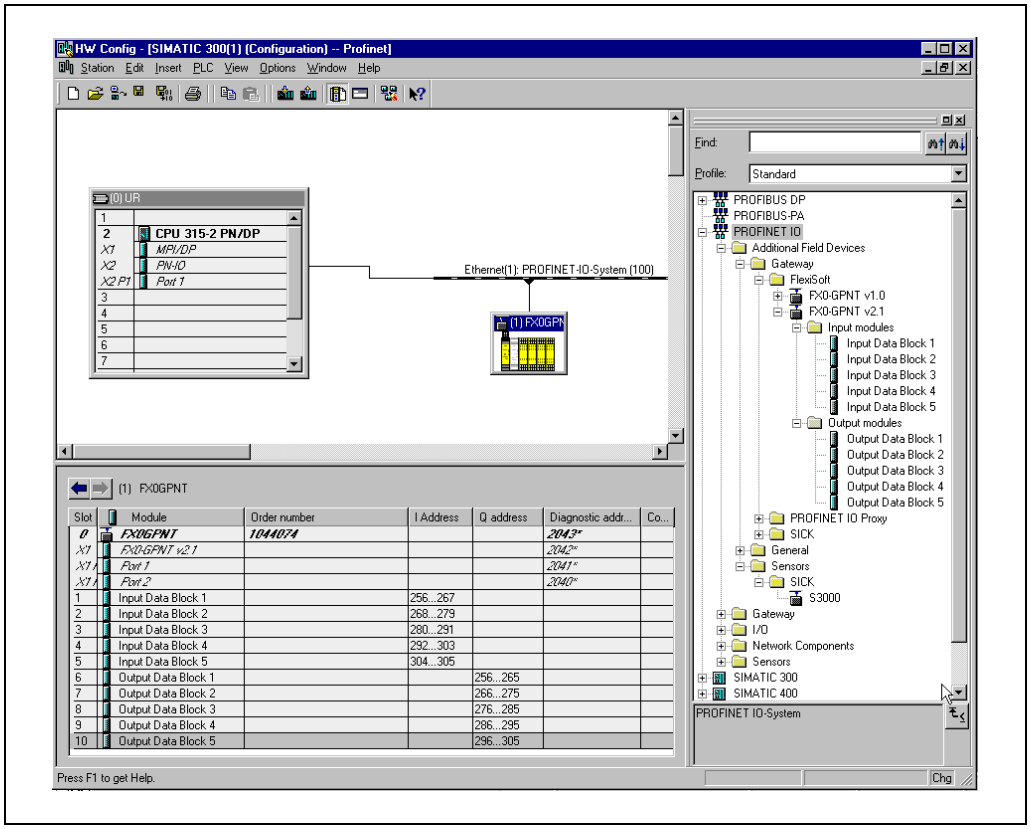


Fig. 42: PROFINET IO gateway in the PROFINET IO HW Config

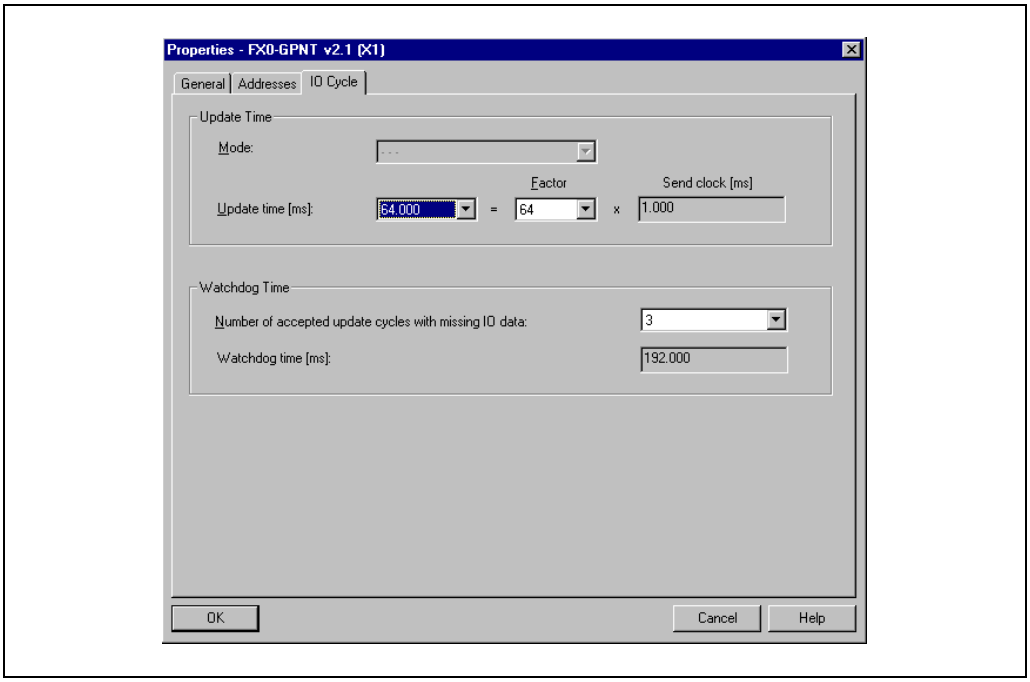
After adding the device to the automation network it is required to configure which of the cyclic data blocks will be used and where they will be addressed in memory. For details refer to section 5.4.4 “PROFINET configuration of the gateway — Which data are transferred” on page 92.

Step 3: Configure the properties of the gateway

➢ Double click on the gateway hardware symbol.

➢ Configure the update time of the cyclic I/O data exchange. To do this click on the **IO cycle** tab and select the desired rate from the **Update time** pull-down menu.

Fig. 43: Configuration of the update time of the FX0-GPNT



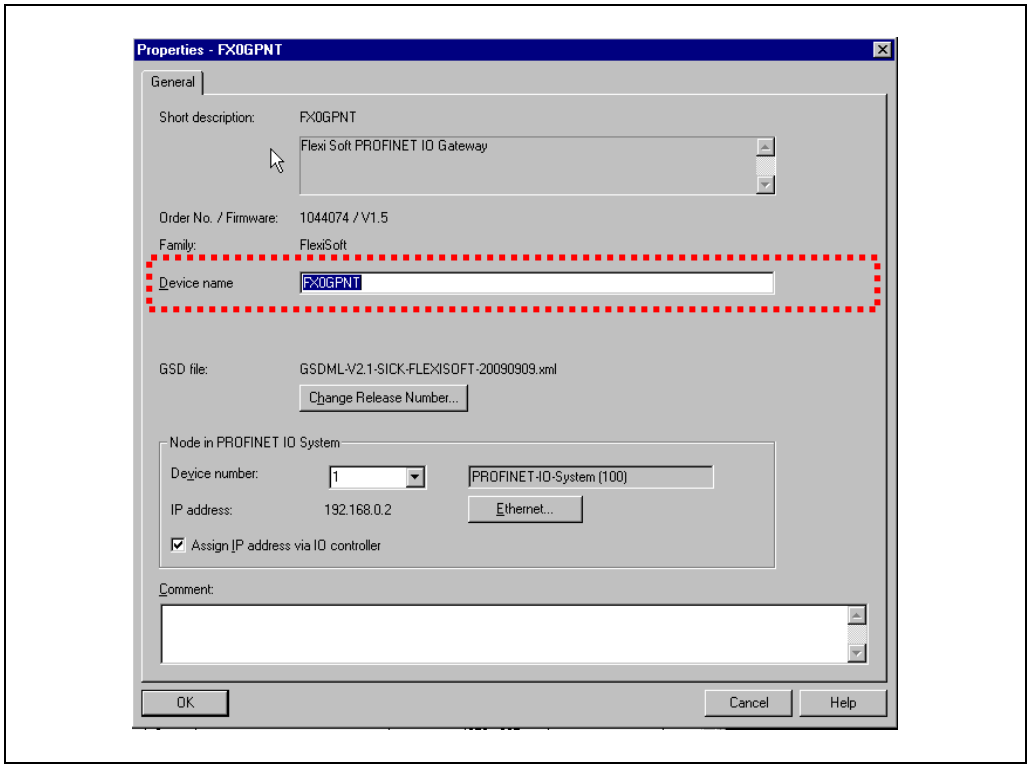
Step 4: Assign the device name

In order for the PLC to communicate with the FX0-GPNT, the PLC software and the gateway must agree on the name of the gateway.

Specify the gateway's PROFINET IO device name

- Double click on the gateway hardware symbol.
- Select the **General** tab.
- Enter the desired device name in the dialog as shown below:

Fig. 44: Properties dialog of the FX0-GPNT

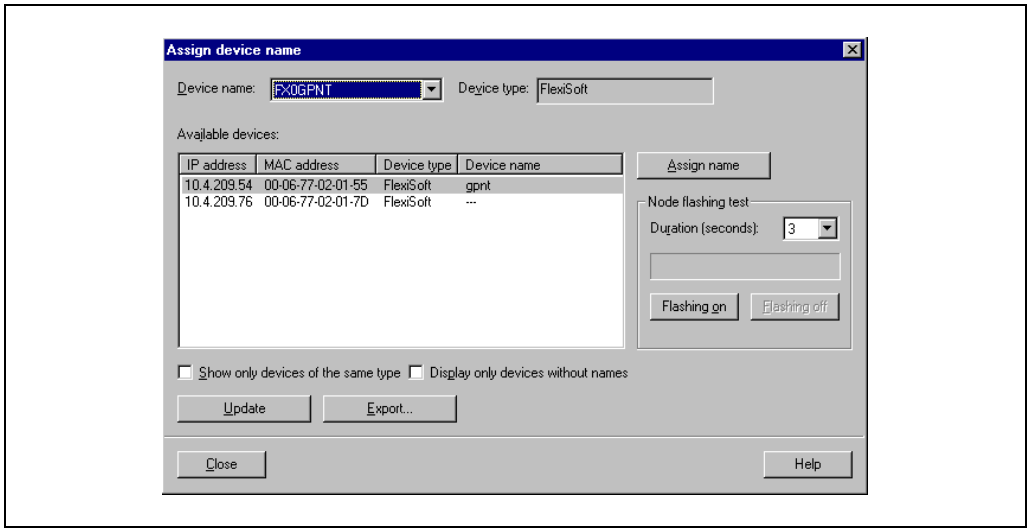


Note The device name format shall apply to the PROFINET standard specification.

Assign the device name to the gateway.

- Select **PLC** > **Ethernet** > **Assign device name**. The **Assign device name** dialog opens.
- From the **Assign device name** dialog, find and select the SICK gateway that you wish to assign the device name to in the list.
- Click the **Assign name** button.

Fig. 45: Assign device name dialog for the FX0-GPNT



5.4.4 PROFINET configuration of the gateway — Which data are transferred

Cyclic data

The physical Flexi Soft I/O modules are not represented as typical hardware modules in the PROFINET IO hardware catalogue. Instead, the data available from the Flexi Soft system has been organized into data blocks. Each data block represents a “hardware” module in the PROFINET IO hardware catalogue. The Flexi Soft PROFINET IO gateway GSDML supports ten (10) slots (see Fig. 46: Projecting the FX0-GPNT) where the modules can be placed into. This allows each data block to be mapped once.

Process data from the Flexi Soft system to the PLC

The FX0-GPNT provides 5 input data blocks (virtual I/O device modules) containing the process image. These must be projected in a hardware configurator (e.g. SIEMENS HW Config) in natural order (1, 2, 3, 4, 5). No other sequence is possible.

Notes

- Depending on the type of PLC used, further modules may be displayed (e.g.“universal module”). These modules are not needed and should be ignored.
- The input data blocks 1-4 contain 12 bytes each, input data block 5 contains 2 bytes.
- The contents of the input data blocks are freely selectable, but are preconfigured in the Flexi Soft Designer configuration software:

Tab. 67: Default content of input data block 1-5 of the FX0-GPNT

	Data block 1	Data block 2	Data block 3	Data block 4	Data block 5
	Input data	Input data	Input data	Input data	Input data
Byte 0	Input values module 1	Output values module 1	Logic result 0	Not assigned	Not assigned
Byte 1	Input values module 2	Output values module 2	Logic result 1	Not assigned	Not assigned
Byte 2	Input values module 3	Output values module 3	Logic result 2	Not assigned	Not available
Byte 3	Input values module 4	Output values module 4	Logic result 3	Not assigned	
Byte 4	Input values module 5	Output values module 5	Gateway direct output values 0	Not assigned	
Byte 5	Input values module 6	Output values module 6	Gateway direct output values 1	Not assigned	
Byte 6	Input values module 7	Output values module 7	Gateway direct output values 2	Not assigned	
Byte 7	Input values module 8	Output values module 8	Gateway direct output values 3	Not assigned	
Byte 8	Input values module 9	Output values module 9	Not assigned	Not assigned	
Byte 9	Input values module 10	Output values module 10	Not assigned	Not assigned	
Byte 10	Input values module 11	Output values module 11	Not assigned	Not assigned	
Byte 11	Input values module 12	Output values module 12	Not assigned	Not assigned	
Length	12 bytes	12 bytes	12 bytes	12 bytes	2 bytes

For detailed information about the content of the process image please see section 3.3 “Data transmitted into the network (input data sets)” on page 14.

For information on how to configure the process image, see chapter 7 “Layout and content of the process image” on page 173 and the Flexi Soft Designer operating instructions (SICK part no. 8012998).

Data from the PLC to the FX3-CPUx

There are five (5) output data blocks, 10 bytes each.

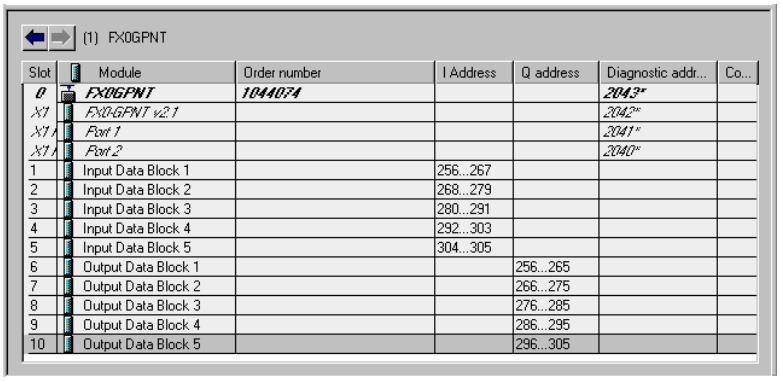
The content of these data blocks can be used as input in the Flexi Soft logic editor or can be routed via a second gateway into another network. In order to have the desired bits available for routing or in the logic editor, tag names have to be defined for each bit that shall be used. Bits without a tag name will not be available.

For detailed information how to define and customize the content and tag names of the input and output data please see chapter 7 “Layout and content of the process image” on page 173 and the operating instructions for the Flexi Soft Designer software (SICK part no. 8012998).

Settings within the PROFINET IO network configuration tool

➢ Drag the data blocks from the SIEMENS SIMATIC Manager — HW Config hardware catalogue under >>**PROFINET IO** > **Additional field devices** > **Gateway** > **SICK** > **Flexi Soft...** > **data blocks** into the slots of the FX0-GPNT shown in the SIEMENS SIMATIC Manager — HW Config configuration table.

Fig. 46: Projecting the FX0-GPNT



Notes

- The I and Q addresses reflect where in memory the cyclic data will be available.
- Each input data block can only be placed into the slot of the same number. Example: Input data block **4** can only be mapped into slot **4**.

Acyclic data and alarms

Record read

Flexi Soft system diagnostic data is available as data record to be read by the PLC. There are three data sets, data set 2, 3 and 4, providing diagnostic information:

- Data Set 2 contains the system CRCs.
- Data Set 3 contains the individual module status with four (4) bytes per module.
- Data Set 4 is currently filled with reserved values.

The format of the data sets is as specified in the tables below.

To access the acyclic data sets, a record read must be performed on the appropriate address as shown in the following table.

Tab. 68: Memory location for data set 2, 3 and 4

	Data set 2	Data set 3	Data set 4
Location	1200-1231	1300-1359	1400-1459
Size in bytes	32 bytes	60 bytes	60 bytes

Note Data set 1 is mapped into the cyclic transferred PROFINET modules of the device. The content may be defined by the user. Refer to chapter 7 “Layout and content of the process image” on page 173 for details.

Tab. 69: Default content of input data set 2-4 of the FX0-GPNT.
For the interpretation of the module status bit in data set 3 please see section 3.3.6 "Error and status information of the modules" on page 19.

	Data set 2	Data set 3	Data set 4
Byte 0	Overall CRC	Module status module 0	Reserved
Byte 1			
Byte 2			
Byte 3			
Byte 4	System CRC (SCID)	Module status module 1	
Byte 5			
Byte 6			
Byte 7			
Byte 8	Reserved	Module status module 2	
Byte 9			
Byte 10			
Byte 11			
Byte 12		Module status module 3	
Byte 13			
Byte 14			
Byte 15			
Byte 16		Module status module 4	
Byte 17			
Byte 18			
Byte 19			
Byte 20		Module status module 5	
Byte 21			
Byte 22			
Byte 23			
Byte 24		Module status module 6	
Byte 25			
Byte 26			
Byte 27			
Byte 28		Module status module 7	
Byte 29			
Byte 30			
Byte 31			
Byte ...		...	
Byte 49		...	
Byte ...		...	
Byte 56		Module status module 14. Module 13 and 14 are always the gateways.	
Byte 57			
Byte 58			
Byte 59			
Length	**32 bytes**	**60 bytes**	**60 bytes**

I&M Information

The FX0-GPNT will support I&M information as required by the PROFINET specification. The following I&M information will be able to be read from the device:

Tab. 70: I&M information of the FX0-GPNT

I&M field	Size	Value
Manufacturer ID	2 bytes	257
Order ID	20 bytes	"1044078 " (must be padded with spaces)
Serial Number	16 bytes	Read from I2C
Hardware Revision	4 bytes	Read from I2C
Software Revision	4 bytes	Read from firmware
Revision Counter	2 bytes	0
Profile ID	2 bytes	Generic Device
Profile Specific Device	2 bytes	Generic Device
IM version	2 bytes	1.1
IM Supported	2 bytes	0

Alarms

Alarms can be read acyclically through the PROFINET IO alarms infrastructure. Once an error occurs on any Flexi Soft module, the PROFINET IO gateway raises the appropriate diagnostic alarm to the network. This will trigger the fault LED on the PLC, and make the specifics (text and help) of the diagnostic alarm available through the SIMATIC PLC interface. The RALRM function block (SFB54) in OB82 (the diagnostic interrupt) allows the user to retrieve specifics on the alarm raised within the PLC program itself.

Notes

- All alarms are output to module 0.
- The subslot number indicates the Flexi Soft module that has caused the alarm. Number 0 = CPU, 1 = 1st XT module, 2 = 2nd XT module ... 13 = 1st gateway, 14 = 2nd gateway.
- The reason for the alarm is being identified by an error text message from the GSDML file. Up to 32 different error messages per Flexi Soft module type are possible.
- For the possible causes for an alarm please refer to Tab. 71.
- The same diagnostic information is available through a record read to data set 3.

The following table matches the PROFINET IO error type (as defined by the GSDML) to the appropriate message.

The module status bits have the following significance, if not otherwise stated:

0 = Error

1 = No error

Tab. 71: PROFINET IO error type definitions

Error type	Message	
	Error origin	**Error definition**
0100	CPU	Operating state Run
0101		Internal tests
0102		Summary of bits 0105 to 0107
0103		Reserved
0104		Configuration Flexi Soft system
0105		Power supply
0106		EFI1
0107		EFI2
0108		Flexi Link stations in the system 1 = All found 0 = One or more are missing
0109		Flexi Link stations suspended 1 = None 0 = One or more
0200	XTIO/XTDI	Operating state Run
0201		Internal tests
0202		Summary of bits 0205 to 0207
0203		Reserved
0204		Configuration of this module is valid
0205		Outputs power supply
0206		Output *Fast-Shut-Off*
0207		Reserved
0208		Input 1-2 dual channel input evaluation
0209		Input 3-4 dual channel input evaluation
0210		Input 5-6 dual channel input evaluation
0211		Input 7-8 dual channel input evaluation
0212 ... 0215		Reserved
0216		External test signal Input 1
0217		External test signal Input 2
0218		External test signal Input 3
0219		External test signal Input 4
0220		External test signal Input 5
0221		External test signal Input 6
0222		External test signal Input 7
0223		External test signal Input 8
0224		Short-circuit monitoring output 1: short-circuit to high
0225		Short-circuit monitoring output 1: short-circuit to low
0226		Short-circuit monitoring output 2: short-circuit to high
0227		Short-circuit monitoring output 2: short-circuit to low
0228		Short-circuit monitoring output 3: short-circuit to high
0229		Short-circuit monitoring output 3: short-circuit to low
0230		Short-circuit monitoring output 4: short-circuit to high
0231		Short-circuit monitoring output 4: short-circuit to low
0300	PROFIBUS gateway	Operating state Run
0301		Internal tests
0302		Summary of bits 0305 to 0307 (external error)
0303		Reserved
0304		Configuration of this module is valid
0305		Communication from the network
0306		Communication to the network
0307 ... 0331		Reserved

Error type	Message	
	Error origin	Error definition
0400	CANopen gateway	Operating state Run
0401		Internal tests
0402		Summary of bits 0405 to 0407 (external error)
0403		Reserved
0404		Configuration of this module is valid
0405		Communication from the network
0406		Communication to the network
0407 ... 0431		Reserved
0500	DeviceNet gateway	Operating state Run
0501		Internal tests
0502		Summary of bits 0505 to 0507 (external error)
0503		Reserved
0504		Configuration of this module is valid
0505		Communication from the network
0506		Communication to the network
0507 ... 0531		Reserved
0600	Modbus gateway	Operating state Run
0601		Internal tests
0602		Summary of bits 0605 to 0607 (external error)
0603		Reserved
0604		Configuration of this module is valid
0605		Communication from the network
0606		Communication to the network
0607 ... 0631		Reserved
0700	EtherNet/IP gateway	Operating state Run
0701		Internal tests
0702		Summary of bits 0705 to 0707 (external error)
0703		Reserved
0704		Configuration of this module is valid
0705		Communication from the network
0706		Communication to the network
0707 ... 0731		Reserved
0800	PROFINET gateway	Operating state Run
0801		Internal tests
0802		Summary of bits 0805 to 0807 (external error)
0803		Reserved
0804		Configuration of this module is valid
0805		Communication from the network
0806		Communication to the network
0805 ... 0831		Reserved
0C00	CC-Link gateway	Operating state Run
0C01		Internal tests
0C02		Summary of bits 0C05 to 0C07 (external error)
0C03		Reserved
0C04		Configuration of this module is valid
0C05		Communication from the network
0C06		Communication to the network
0C07 ... 0C31		Reserved
0F00	Sercos III gateway	Operating state Run
0F01		Internal tests
0F02		Summary of bits 0F05 bis 0F07 (external error)
0F03		Reserved
0F04		Configuration of this module is valid
0F05		Communication from the network
0F06		Communication to the network
0F07 ... 0F31		Reserved

Error type	Message	
	Error origin	Error definition
1000	EtherCAT gateway	Operating state Run
1001		Internal tests
1002		Summary of bits 1005 to 1007 (external error)
1003		Reserved
1004		Configuration of this module is valid
1005		Communication from the network
1006		Communication to the network
1007 ... 1031		Reserved
1100 ... 1F31	Other gateways	Reserved
2000	STIO	Operating state Run
2001		Internal tests
2002		Summary of bits 2005 to 2007
2003		Reserved
2004		Configuration of this module is valid
2005		Outputs power supply
2006		Reserved
2007		Output load (overcurrent) monitoring
2008 ... 2031		Reserved
2100	MOCx	Operating state Run
2101		Internal tests
2102		Summary of bits 2105 to 2107
2103		Reserved
2104		Configuration of this module is valid
2105		Encoder 1
2106		Encoder 2
2107		Reserved
2108 ... 2111		Reserved
2112		User-defined status bit 1[12)]
2113		User-defined status bit 2[12)]
2114		User-defined status bit 3[12)]
2115		User-defined status bit 4[12)]
2116 ... 2131		Reserved
2200	XTDS	Operating state Run
2201		Internal tests
2202		Summary of bits 2205 to 2207
2203		Reserved
2204		Configuration of this module is valid
2205		Outputs power supply
2206		Reserved
2207		Output load (overcurrent) monitoring
2208		Input 1-2 dual channel input evaluation
2209		Input 3-4 dual channel input evaluation
2210		Input 5-6 dual channel input evaluation
2211		Input 7-8 dual channel input evaluation
2212 ... 2215		Reserved
2216		External test signal Input 1
2217		External test signal Input 2
2218		External test signal Input 3
2219		External test signal Input 4
2220		External test signal Input 5
2221		External test signal Input 6
2222		External test signal Input 7
2223		External test signal Input 8
2224 ... 2231		Reserved
2300 ... 3F31	Other modules	Reserved

[12)] The status of this bit can be defined to suit the specific application in the MOCx logic, e.g. to indicate inadmissible movements of an axis that have been detected by an MOCx function block.

5.4.5 TCP/IP configuration interface

See section 5.1.1 “TCP/IP configuration interface” on page 32.

5.4.6 Ethernet TCP/IP socket interface

See section 5.1.2 “Ethernet TCP/IP socket interface” on page 37.

5.4.7 Diagnostics and troubleshooting

For information how to perform diagnostics on the Flexi Soft system please refer to the operating instructions for the Flexi Soft Designer software (SICK part no. 8012998).

Tab. 72: Troubleshooting for the FX0-GPNT

Symbol description:
○: LED is off
● Green: LED lights up green
☼ Red: LED flashes red

Error	Possible cause	Possible remedy
The Flexi Soft Designer tool does not connect to the Flexi Soft gateway module.	FX0-GPNT has no power supply. FX0-GPNT is not in the same physical network as the PC. The PC is configured to another subnet mask in the TCP/IP settings. FX0-GPNT has already been configured once and has a fixed set IP address or an IP address assigned by a DHCP server that is not recognised.	Establish the power supply. Check the Ethernet wiring and network settings on the PC and correct if necessary. Set the subnet mask on the PC to 255.255.0.0 (factory setting of the FX0-GPNT). Check the communication settings in the Flexi Soft Designer.
FX0-GPNT does not supply any data. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Red/green[13)]	FX0-GPNT is configured for data transfer to PLC, but Ethernet communication is not yet established or faulty. Duplicate IP address detected. Another device on the network has the same IP address. Improperly formatted PROFINET device name.	Minimum one Ethernet connection needs to be established. Set up Ethernet connection on PLC side, check Ethernet cabling, check Ethernet connection settings on PLC and in the Flexi Soft Designer. If no Ethernet communication is required, disable the Ethernet connections/PLC interfaces on the FX0-GPNT. Adjust IP address and power cycle device. Adjust device name between PROFINET Master and FX0-GPNT.
FX0-GPNT does not supply any data. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Red (1 Hz)	Configuration required. Configuration download is not completed.	Configure the FX0-GPNT and download the configuration to the device. Wait until the configuration download has been completed.
FX0-GPNT does not supply any data. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Green (1 Hz)	No data set is activated. Flexi Soft system is in Idle mode.	Activate at least one data set. Start CPU (change into Run state).
FX0-GPNT does not supply any data. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Green (2 Hz)	PROFINET master requested LED flashing for physical device identification.	Stop flashing with Simatic Manager or power cycle Flexi Soft system to clear.
FX0-GPNT functioned correctly after configuration, but suddenly no longer supplies data. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Red/green[13)]	FX0-GPNT is operated in slave mode, the IP address is assigned from a DHCP server. After the FX0-GPNT or the DHCP server has been restarted, a different IP address that is unknown to the PLC has been assigned to the FX0-GPNT.	Either assign a fixed IP address to the FX0-GPNT, or reserve a fixed IP address for the FX0-GPNT in the DHCP server (manual assignment by means of the MAC address of the FX0-GPNT).
FX0-GPNT/Flexi Soft system is in Critical fault mode. PWR ● Green LINK/ACT ☼ Green STATUS ● Red	FX0-GPNT is not plugged properly into the other Flexi Soft module. Module connecting plug is soiled or damaged. Other Flexi Soft module has internal critical error.	Plug the FX0-GPNT in correctly Clean the connecting socket/plug. Repower the system. Check the other Flexi Soft modules.
FX0-GPNT is in Critical fault mode. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Red (2 Hz)	FX0-GPNT internal device error CPU firmware version does not support Flexi Soft gateways.	Switch off the power supply of the Flexi Soft system and switch it on again. Check the diagnostics messages with Flexi Soft Designer. Use a FX3-CPUx with the required firmware version (see section 2.2 “Correct use” on page 10). If the error remains, replace the gateway.

[13)] With firmware ≥ V2.00.0 the red/green flashing of the STATUS LED can be disabled in the Flexi Soft Designer. In this case, the STATUS LED is ● Green permanently, if the configuration is valid.

5.5 EtherCAT gateway

"EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany."

The following Flexi Soft gateway can be used for EtherCAT: FX0-GETC.

You will find the ESI file and the device icon for PLC interfacing with EtherCAT support ...

- in the Internet on the FX0-GETC product page on www.sick.com.
- in the Flexi Soft Designer program folder on your hard disk (default installation folder is "C:\programs\SICK\FlexiSoft\DeviceDescriptions\...").

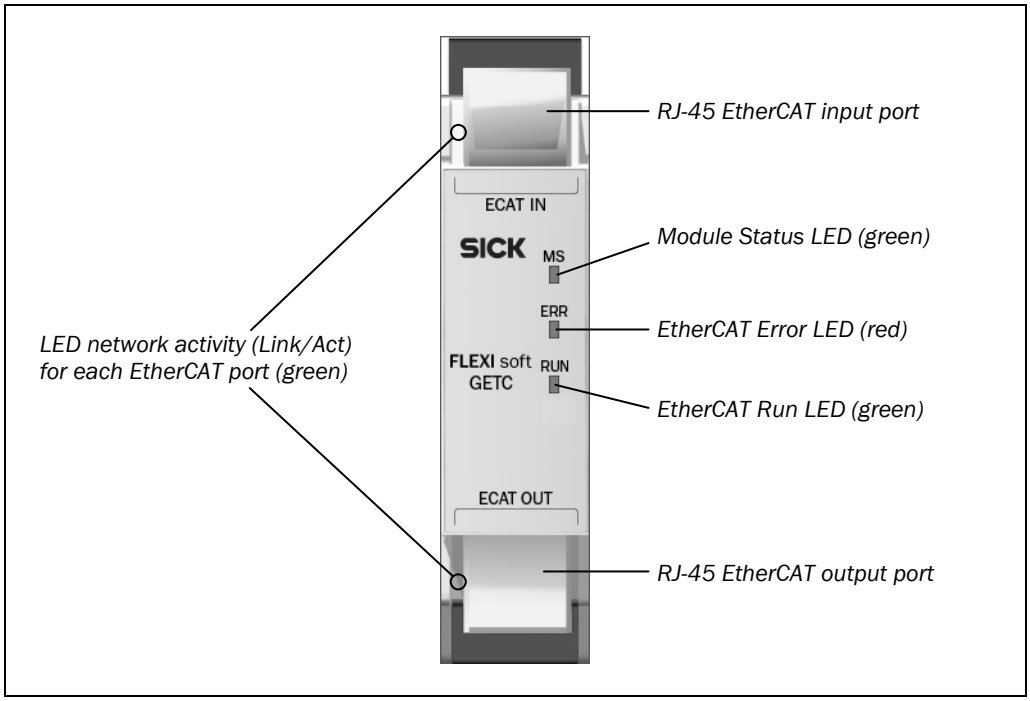
The FX0-GETC is an EtherCAT slave device. It supports and requires the following services for complete function:

- Configuration and diagnostics of the Flexi Soft station via TCP/IP, tunneled in EtherCAT using the EoE (Ethernet over EtherCAT) protocol
- CoE (CAN application layer over EtherCAT)
- Station diagnostics via CoE object 10F3h

5.5.1 Interfaces and operation

Controls and status indicators

Fig. 47: Controls and status indicators FX0-GETC



Tab. 73: Meaning of the status LEDs of the FX0-GETC

Symbol description:
○: LED is off
● Green: LED lights up green
☼● Red: LED flashes red

LED		Meaning
MS	○	Power-up
	● Green	Flexi Soft system is in the Run state.
	☼● Green (1 Hz)	Flexi Soft system is in the Stop state.
	☼● Red (1 Hz)	Invalid configuration
	☼● Red (2 Hz)	Critical error on the gateway
	● Red	Critical fault on another module
	☼● Red/green	Recoverable external error
ERR	○	No error
	☼● Red (2.5 Hz)	Invalid configuration
	☼● Red (single flash)	Pre-operational caused by system behaviour (e.g. configuration in progress or required)
	☼● Red (double flash)	Timeout (connection lost)
	● Red	System error
RUN	○	Init
	☼● Green (2.5 Hz)	Pre-operational
	☼● Green (single flash)	Safe-operational
	● Green	Operational
Link/ Act	○	No EtherCAT connection
	● Green	EtherCAT connection active, no data transmission
	☼● Green	EtherCAT connection active, data transmission

Note For error diagnostics and troubleshooting see section 5.5.10 “Diagnostics and troubleshooting” on page 114.

Power-up sequence

On power-up, the following LED test sequence is performed:

- All LEDs ○ Off for 6 s
- MS LED ● Red for 0.25 s
- MS LED ● Green for 0.25 s
- MS LED ○ Off
- ERR LED ● Red for 0.25 s
- ERR LED ○ Off
- RUN LED ● Green for 0.25 s
- RUN LED ○ Off

Cabling requirements

EtherCAT is based on a Fast Ethernet cabling infrastructure that is characterized as follows:

- type 100 Base-TX
- RJ-45 connectors
- twisted pair Ethernet cable, max. length 100 m according to DIN EN 50 173
- use of core pairs 1/2 and 3/6
- screened cables are recommended

5.5.2 Installation of the gateway in the Flexi Soft system

This section outlines the basic steps to install the gateway in the Flexi Soft system. More detailed information will be given in the following sections.

Add the gateway to a Flexi Soft system

- Mount the gateway as described in section 4.1.1 “Steps for mounting the modules” on page 24.
- Connect the FX0-GETC to your EtherCAT network using Ethernet cables with RJ-45 connectors. The upper connector on the gateway is the input port (ECAT IN) and is connected to the PLC side of the network. The lower connector is the output port (ECAT OUT) where further EtherCAT devices can be connected. If the output port is not used it can remain unconnected.
- Open the Flexi Soft Designer and load the hardware configuration including the gateway or create a new Flexi Soft system with an FX0-GETC in the Flexi Soft Designer.

Configure the gateway in the Flexi Soft system

- Click on the **Gateways** button above the main window and select the FX0-GETC or double click the FX0-GETC in the hardware configuration view to open the gateway configuration dialog. The configuration dialog is divided into three areas: **CPU to EtherCAT**, **EtherCAT to CPU** and **Gateway configuration** that can be accessed through the icons on the left.
- In the **CPU to EtherCAT** area, select the data that shall be transferred from the Flexi Soft system to the EtherCAT network. Up to 50 bytes can be used which are divided in five input data sets with 10 bytes each.
- In the **EtherCAT to CPU** area, select the data that shall be transferred from the EtherCAT network to the Flexi Soft system. Up to 50 bytes can be used which are divided in five output data sets with 10 bytes each.
- In the **Gateway configuration** area, you can change the device name of the gateway in the Flexi Soft system. The default name of the gateway is “GETC”.

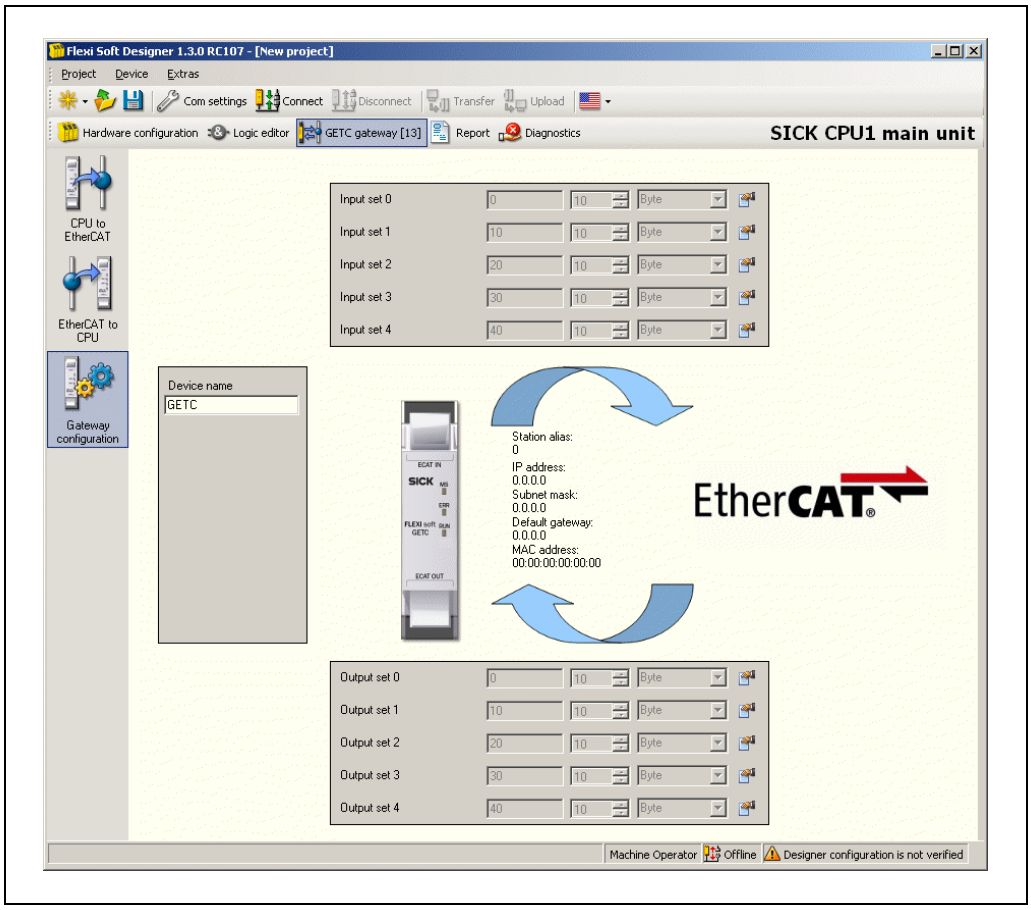


Fig. 48: Gateway configuration dialog for the FX0-GETC

Note In the middle of the screen, the station alias (second station address) of the gateway and the EoE parameters are displayed. This data is only informative; it is set in the EtherCAT network configuration tool (e.g. TwinCAT).

5.5.3 EtherCAT configuration of the gateway

Note This document does not cover the creation of the EtherCAT network or the rest of the automation system project in the network configuration tool. It is assumed the EtherCAT project has already been set up in the configuration program, e.g. TwinCAT. Examples refer to configurations performed with TwinCAT V2.11.0.

The following steps need to be taken in order to configure the communication between PLC and gateway.

Step 1: Install the EtherCAT Slave Information (ESI) file

The ESI file *FX0-GETC.xml* contains the information that is needed for the integration of the FX0-GETC in the EtherCAT network. Before the FX0-GETC can be used as device in the EtherCAT network configuration tool (e.g. TwinCAT) for the first time, the ESI file of the gateway must be installed into the hardware catalogue of the tool.

- ➢ You will find the ESI file in the Flexi Soft Designer program folder (default folder is “C:\programs\SICK\FlexiSoft\DeviceDescriptions\FX0-GETC_ESI”).
- ➢ You can also download the ESI file from www.sick.com, on the FX0-GETC product page.
- ➢ Follow the instructions in the online help or in the user manual of the EtherCAT network configuration tool for installing ESI files.

Example — How to install the ESI file using TwinCAT:

- ➢ Copy the ESI file FX0-GETC.xml in the TwinCAT folder under “TwinCAT\Io\EtherCAT\”.
- ➢ Restart TwinCAT in order to rebuild its ESI cache.

Step 2: Insert the gateway into the EtherCAT network

In order to have the Flexi Soft system data available in the PLC process image, the gateway must be added to the hardware configuration first. The procedure associated with this depends on the hardware configuration program of the PLC being used. On this topic, please also read the documentation for the corresponding program.

Example — How to insert the FX0-GETC using TwinCAT:

➢ In order to integrate the gateway manually into the EtherCAT network, use the **Append box** command and select the Flexi Soft EtherCAT gateway.

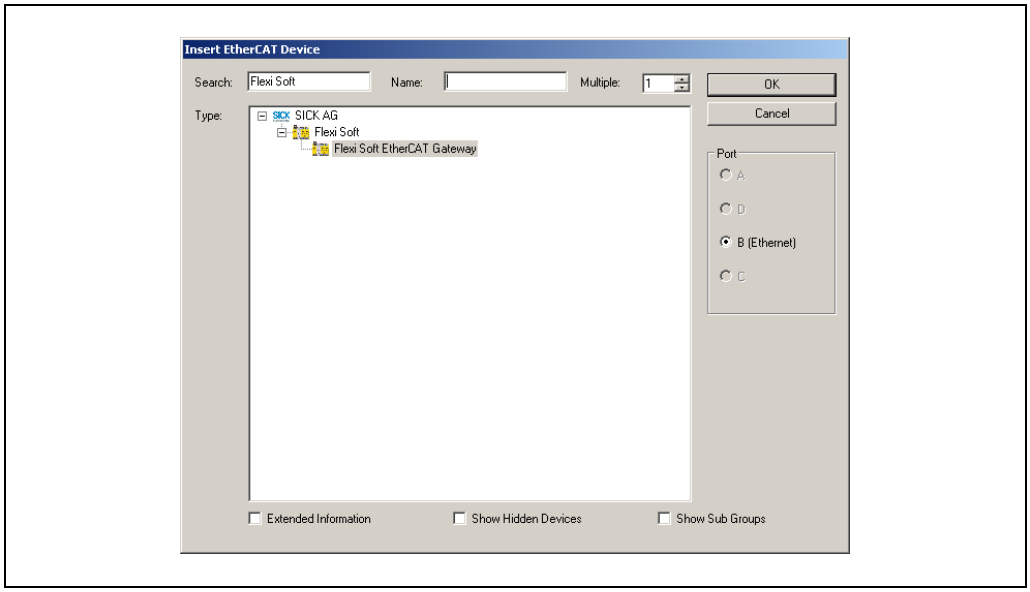


Fig. 49: Example for inserting the FX0-GETC into an EtherCAT network

➢ Alternatively you can perform a network scan using the **Scan boxes** command.

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

After adding the device to the automation network you must configure which of the process data objects (PDOs) you want to use.

The FX0-GETC provides five input PDOs for the transfer of input data to a connected PLC that can be used alternatively. I.e. only one of these five input PDOs can be active at any time. There is one input PDO for 10 byte input data (1 data set used in the Flexi Soft Designer), one for 20 byte data (2 data sets used) etc. up to the maximum of 50 byte. Similarly you must select one out of five different available output PDOs from 10 to 50 bytes into which the PLC output data can be written.

Notes

- The structure of the PDOs is predefined and can not be edited.
- The input data PDOs contain an additional first byte for the diagnostics flag (*Diag*). This byte is set to True (“1”) if a new diagnosis message (CoE object 10F3h) is available and to False (“0”) if all diagnosis messages are acknowledged.

Tab. 74: Process data objects of the FX0-GETC

Input PDOs		
Index	**Size**	**Content**
1A00h	11 bytes	Diag byte + input data set 1
1A01h	21 bytes	Diag byte + input data sets 1-2
1A02h	31 bytes	Diag byte + input data sets 1-3
1A03h	41 bytes	Diag byte + input data sets 1-4
1A04h	51 bytes	Diag byte + input data sets 1-5
Output PDOs		
Index	**Size**	**Content**
1600h	10 bytes	Output data set 1
1601h	20 bytes	Output data sets 1-2
1602h	30 bytes	Output data sets 1-3
1603h	40 bytes	Output data sets 1-4
1604h	50 bytes	Output data sets 1-5

➢ Select one of the five available PDOs for each data transfer direction (input and output) with the appropriate size for the used process data in the EtherCAT network configuration tool.

Notes

- Only one input PDO and one output PDO can be active at any time.
- If the selected PDO is greater than the process data configured, unused data will be filled with zeroes.
- If the selected PDO is smaller than the process data configured, excess data will be truncated.

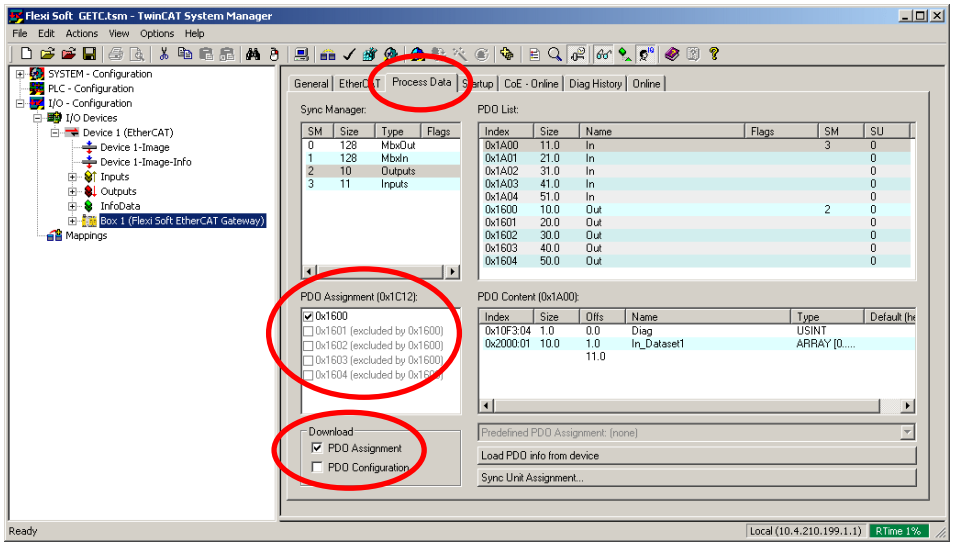
Example — How to select the PDOs using TwinCAT:

➢ Under the **Process Data** file card, select the desired PDO type (**Inputs** or **Outputs**) in the **Sync Manager** selection list.

➢ Then select the desired PDO in the **PDO Assignment** selection list. In order to change the PDO you need to deselect the active one first.

➢ In the **Download** area, select **PDO Assignment** but leave **PDO Configuration** unchecked since the PDO configuration is predefined and can not be changed.

Fig. 50: PDO configuration in the EtherCAT network configuration tool



5.5.4 Input data — Flexi Soft to EtherCAT

The FX0-GETC can transfer up to 50 bytes input data to a connected PLC over EtherCAT. The input data are divided into five data sets.

Notes

- Each input data set contains 10 bytes.
- The contents of the data sets are freely selectable, but are preconfigured in the Flexi Soft Designer configuration software (see Tab. 75).
- If an input data set contains any data, the complete 10 byte set is sent over EtherCAT.

Tab. 75: Default content of input data set 1-5 of the FX0-GETC

	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5
	Input data	**Input data**	**Input data**	**Input data**	**Input data**
Byte 0	Input values module 1	Input values module 11	Output values module 9	Gateway direct output values 2	Not assigned
Byte 1	Input values module 2	Input values module 12	Output values module 10	Gateway direct output values 3	Not assigned
Byte 2	Input values module 3	Output values module 1	Output values module 11	Not assigned	Not assigned
Byte 3	Input values module 4	Output values module 2	Output values module 12	Not assigned	Not assigned
Byte 4	Input values module 5	Output values module 3	Logic result 0	Not assigned	Not assigned
Byte 5	Input values module 6	Output values module 4	Logic result 1	Not assigned	Not assigned
Byte 6	Input values module 7	Output values module 5	Logic result 2	Not assigned	Not assigned
Byte 7	Input values module 8	Output values module 6	Logic result 3	Not assigned	Not assigned
Byte 8	Input values module 9	Output values module 7	Gateway direct output values 0	Not assigned	Not assigned
Byte 9	Input values module 10	Output values module 8	Gateway direct output values 1	Not assigned	Not assigned
Length	10 bytes	10 bytes	10 bytes	10 bytes	10 bytes

For detailed information about the content of the process image please see section 3.3 “Data transmitted into the network (input data sets)” on page 14.

For further information on how to configure the process image, see chapter 7 “Layout and content of the process image” on page 173 and the Flexi Soft Designer operating instructions (SICK part no. 8012998).

Note The process data can also be read using the CoE objects 2000h and 2001h (see section 5.5.9 “CoE (CAN application layer over EtherCAT)” on page 111). The simple SDO access is recommended for diagnostic purposes. In normal operation the faster PDO communication should be used.

5.5.5 Output data — EtherCAT to Flexi Soft

The FX0-GETC can receive up to 50 bytes input data from a connected PLC over EtherCAT. Similar to the input data, the output data are divided into five data sets.

Notes

- Each output data set contains 10 bytes.
- The contents of the output data sets can be configured in the Flexi Soft Designer configuration software. See section 7.3.6 “Output data configuration (Network to Flexi Soft)” on page 180.

5.5.6 Tag name export

The Flexi Soft Designer allows you to export the names of the bits used in the input and output data sets. Before exporting you can also edit the start addresses for the used data sets in the PLC. The exported tag names and start addresses can then be imported as variables to the application program in the EtherCAT network configuration tool (e.g. TwinCAT PLC). This speeds up the generation of the PLC program and makes it easier to identify each bit in the EtherCAT PDOs.

How to export the tag names:

- The input and output data sets must be exported separately. Depending on whether you want to export the tag names of the input or output data sets, open the **CPU to EtherCAT** or the **EtherCAT to CPU** configuration page.
- Click on the **Export** button in the toolbar. A dialog window opens.
- Select the destination folder, enter a name for the export file, select the required file type (e.g. *.csv or *.exp for TwinCAT) from the selection list at the bottom of the dialog window and click on **Save** to export the file.

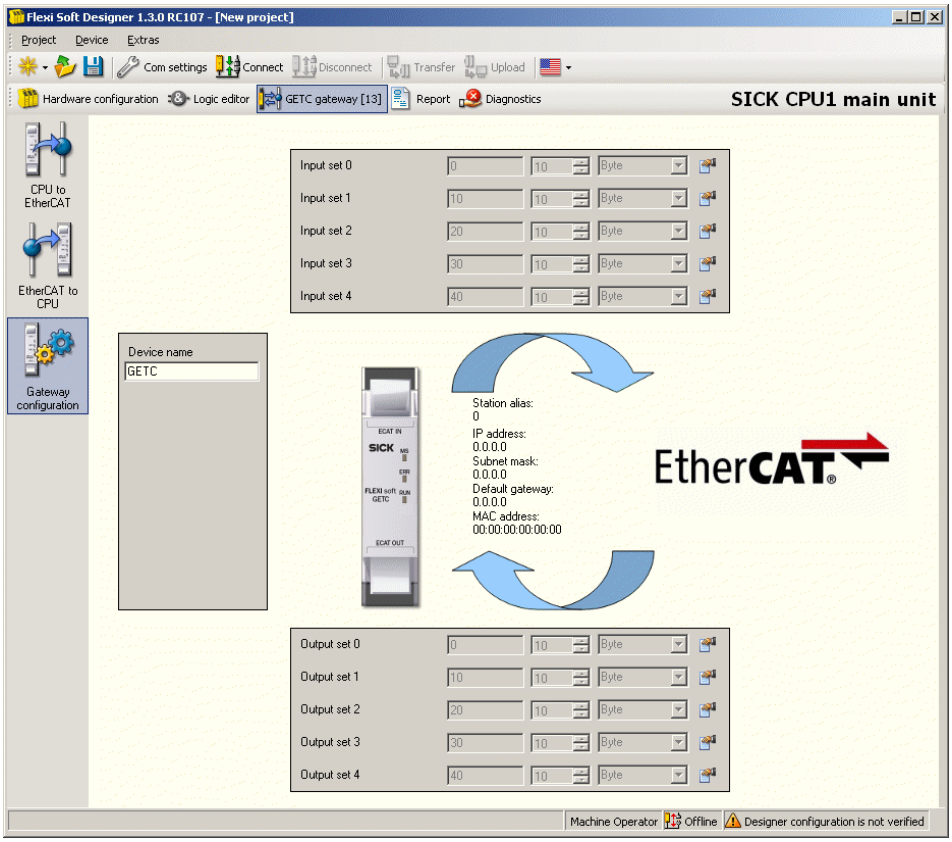
Notes

- For each used data set the export creates a 10 byte structure with the byte or module names and a bit variable for each used bit.
- The name of the bit variable consists of the application name, the name of the byte and the name of the bit.
- It is recommended to assign a tag name to each module in the Flexi Soft configuration and to use unique tag names for all modules, bytes and bits. Special characters in the names are deleted and spaces are replaced with the character “_”.
- The start address for each data set in the TwinCAT PLC process image can be changed in the gateway configuration menu (see below).

How to change the start address for the data sets:

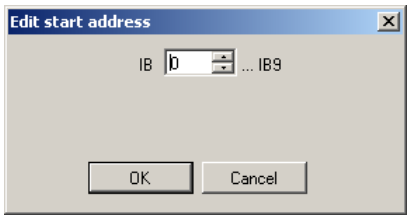
- Open the Flexi Soft Designer and load the hardware configuration including the EtherCAT gateway. Ensure your project is offline.
- Click on the **Gateways** button above the main window and select the FX0-GETC or double click the FX0-GETC in the hardware configuration view to open the gateway configuration dialog.
- Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 51: EtherCAT gateway configuration dialog



- Click on the button to the right of the data set for which you want to change the start address. The following dialog appears:

Fig. 52: Edit the data block start address



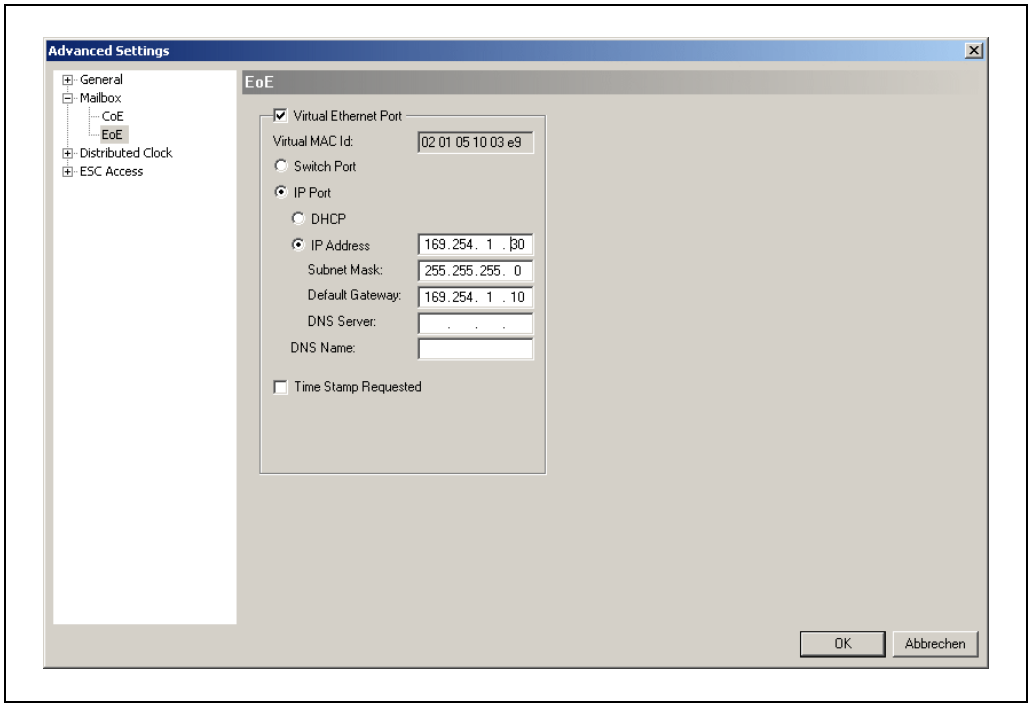
- Type in the desired new start address or use the arrows to change the address. The address will be checked automatically for plausibility, i.e. it is not possible to configure data sets with overlapping address ranges.
- Click **OK** to accept the new start address.

5.5.7 Ethernet over EtherCAT (EoE)

The EoE functionality of the gateway needs to be activated via the EtherCAT network configuration tool (e.g. TwinCAT) before it can be used. The gateway itself has no real MAC address. Therefore it is necessary to assign a virtual MAC address and the IP settings to the device.

➢ Follow the instructions in the online help or in the user manual of your EtherCAT network configuration tool for activating EoE.

Fig. 53: EoE activation for the FX0-GETC in TwinCAT



If the EoE configuration is loaded to the gateway it can be accessed over Ethernet by the Flexi Soft Designer.

Note The EoE protocol only works if the gateway is in the *Pre-Operational* state or higher, because it relies on the EtherCAT mailboxes of the Gateway, which are not available in the *Init* state.

5.5.8 TCP/IP configuration interface

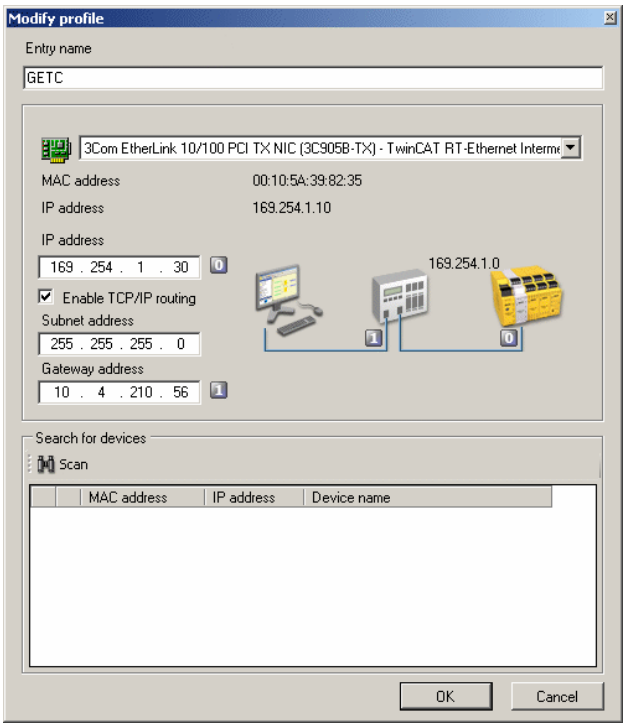
In order to use the TCP/IP communication interface, the FX0-GETC must be part of a working EtherCAT network. EoE functionality must be enabled for the Flexi Soft system and an IP address and a subnet mask must be assigned to it in the EtherCAT configuration tool.

➢ Follow the instructions in the online help or in the user manual of your EtherCAT network configuration tool for assigning an IP address and a subnet mask.

If the EtherCAT master and the Flexi Soft Designer are running on different PCs, then TCP/IP routing must be enabled in the Flexi Soft Designer.

➢ In the **Com settings** window, create a new TCP/IP profile or modify an existing TCP/IP profile for the FX0-GETC, activate the checkbox for **Enable TCP/IP routing** and enter a suitable **Subnet address** and **Gateway address**.

Fig. 54: Enabling TCP/IP routing for the FX0-GETC in the Flexi Soft Designer



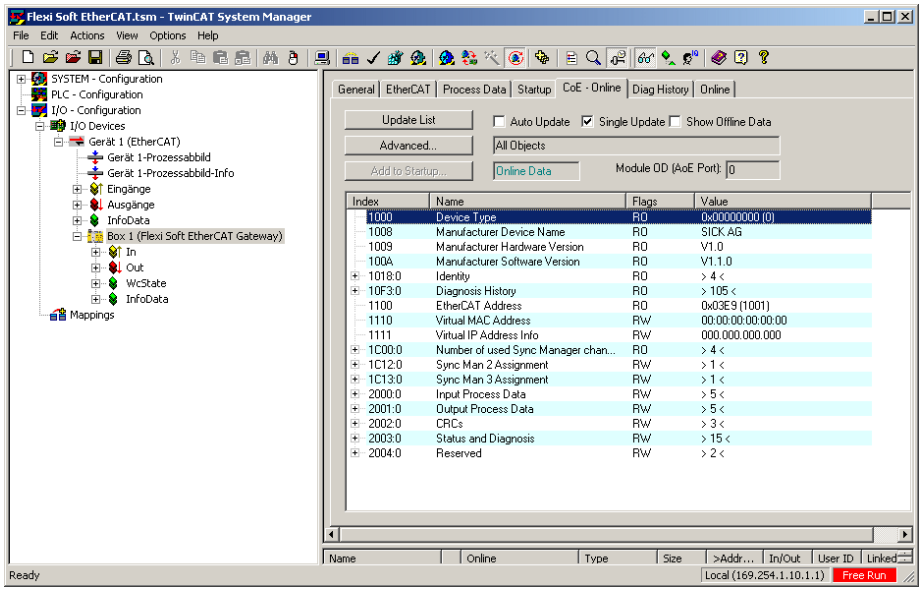
Note The **Default Gateway** in the EoE settings of the EtherCAT gateway in your EtherCAT network configuration tool must be set correctly in order to establish a communication.

5.5.9 CoE (CAN application layer over EtherCAT)

The FX0-GETC supports various CoE objects. They can be viewed in the EtherCAT network configuration tool or used in an application through SDO read commands.

In addition to the standard EtherCAT objects the FX0-GETC has a number of vendor specific objects.

Fig. 55: CoE object catalogue of the FX0-GETC in TwinCAT



Note The CoE objects can only be read, i.e. it is not possible to change the input or output process data or other CoE objects via SDO commands.

Input process data (2000h)

This object contains the input process data from the Flexi Soft system to an EtherCAT PLC and makes them available for acyclic usage. It corresponds to the EtherCAT input PDO data of the FX0-GETC.

Tab. 76: FX0-GETC Input Process Data CoE object 2000h

Index	Subindex	Name	Size
2000h	01	Dataset1	10 bytes
	02	Dataset2	10 bytes
	03	Dataset3	10 bytes
	04	Dataset4	10 bytes
	05	Dataset5	10 bytes

Output process data (2001h)

This object contains the output process data from an EtherCAT PLC to the Flexi Soft system and makes them available for acyclic usage. It corresponds to the EtherCAT output PDO data of the FX0-GETC.

Tab. 77: FX0-GETC Output Process Data CoE object 2001h

Index	Subindex	Name	Size
2001h	01	Dataset1	10 bytes
	02	Dataset2	10 bytes
	03	Dataset3	10 bytes
	04	Dataset4	10 bytes
	05	Dataset5	10 bytes

CRCs (2002h)

This object contains the CRCs described in section 3.3.5 “Configuration checksums (CRCs)” on page 19.

Tab. 78: FX0-GETC CRC CoE object 2002h

Index	Subindex	Name	Size
2002h	01	Overall CRC	4 bytes
	02	System CRC	4 bytes
	03	EFI ACR CRC (reserved for future use)	4 bytes

Status and diagnosis (2003h)

This object contains the module status bits of the Flexi Soft system. Each Flexi Soft module has 32 status bits, each representing a possible error message of the module. The meaning of each bit depends on the type of module. For details see section 3.3.6 “Error and status information of the modules” on page 19.

The gateway uses these module status bits internally in order to generate the error messages displayed in object 10F3h.

Tab. 79: FX0-GETC Status and Diagnosis CoE object 2003h

Index	Subindex	Name	Size
2003h	01	FX3-CPUx	4 bytes
	02	Module 1	4 bytes
	03 ... 0D	Module 2 ... Module 12	4 bytes
	0E	Gateway 1	4 bytes
	0F	Gateway 2	4 bytes

Reserved (2004h)

This object is reserved for future use.

Diagnosis history (10F3h)

The **Diagnosis history** lists the entries in object 2003h chronologically. If the **Diagnosis history** contains new entries that have not yet been confirmed, the *Diag* byte in the input process image (i.e. the first byte of the EtherCAT input PDO and of the CoE object 2000h) is set to True.

Subindex 1 of the **Diagnosis history** contains the maximum number of possible diagnosis history entries. Subindex 2 (Newest) refers to the newest diagnosis message. Subindex 3 (Acknowledged) refers to the last message that has been confirmed or — if no messages have been confirmed so far — to the last entry. Subindex 4 is True if Reading is required (i.e. if Newest and Acknowledged are different).

Tab. 80: Structure of the Diagnosis history object

Subindex [hex]	Content	Format	Comments
01h	Max. entry number	UNSIGNED8	
02h	Newest	UNSIGNED8	= Subindex of the newest history entry (e.g. 2Ah)
03h	Acknowledged	UNSIGNED8	= Subindex of the last acknowledged history entry
04h	Reading required	BOOLEAN	= True, if Newest is not the same as Acknowledged
05h	Flags	UNSIGNED16	Flags to control the sending and storing of diagnosis messages — the Flexi Soft EtherCAT gateway does not support any of the optional options.
06h-69h	Diagnosis history entries	OCTET STRING	See below.

The **Diagnosis history** object is structured as a ring buffer. If subindex 69h has been written, the next entry will start over with subindex 06h.

If the number of unconfirmed diagnosis messages reaches 100, older messages will not be overwritten. Instead the newest diagnosis message is replaced by a buffer overflow error message (FFFFh).

Detailed information about the structure and usage of this object can be found in the “EtherCAT Protocol Enhancements” document by the EtherCAT Technology Group (ETG.1020) which can be obtained via the internet on www.ethercat.org.

Each diagnosis message consists of a diagnosis code and an ASCII string containing the message parameter set.

The diagnosis code consists of the module number and the diagnosis bit of the respective module.

The ASCII string is “module xx +” or “module xx -”, where xx stands for the position in the Flexi Soft system of the module that has generated the diagnosis message. Coming diagnosis messages are marked with a “+”, going diagnosis messages are marked with a “-”.

If a problem has been diagnosed and resolved subsequently, the object 10F3h will contain two diagnosis messages that differ only by the trailing “+” or “-”.

The FX0-GETC does not support a time stamp for the **Diagnosis history** object. If a time stamp is required, the reading device (e.g. the PLC) can be programmed to add the time stamp when a diagnosis message is read.

5.5.10 Diagnostics and troubleshooting

The FX0-GETC provides the following module status bits in the Flexi Soft system for diagnostic purposes. The module number of the FX0-GETC is 16.

Tab. 81: Module status bits of the FX0-GETC

Bit	Error message	Possible causes
0	Module not in operation	No process data connection with the PLC
1	Module has internal error	Hardware error. Replace module
2	Module has external error	No process data connection with the PLC
3	Reserved	n/a
4	Configuration invalid or incompatible	The configuration of the Flexi Soft system is invalid, e.g. a hardware component has been added or removed or has been replaced by an incompatible device
5	Module input status invalid	No process data connection with the PLC Configuration required or in progress EtherCAT status is Pre-operational or less
6	Module output status invalid	No process data connection with the PLC EtherCAT status is Safe-operational or less
7 ... 31	Reserved	n/a

For information how to perform diagnostics on the Flexi Soft system please refer to the operating instructions for the Flexi Soft Designer software (SICK part no. 8012998).

Tab. 82: Troubleshooting for the FX0-GETC

Symbol description:
○: LED is off
● Green: LED lights up green
☼ Red: LED flashes red

Error	Possible cause	Possible remedy
The Flexi Soft Designer tool does not connect to the Flexi Soft gateway module	FX0-GETC has no power supply.	Establish the power supply. Check the communication settings in the Flexi Soft Designer.
FX0-GETC does not supply any data. MS ☼ Red (1 Hz) ERR ☼ Red (2.5 Hz) RUN ○ Off	Configuration required. Configuration download is not completed.	Configure the FX0-GETC and download the configuration to the device. Wait until the configuration download has been completed.
FX0-GETC does not supply any data. MS ● Red/green ERR ○ Off RUN ● Green (2.5 Hz)	No input PDO is activated.	Activate an input PDO.
FX0-GETC does not supply any data. MS ☼ Green (1 Hz) ERR ○ Off RUN ● Green	Flexi Soft system is in the Stop state.	CPU/application is stopped. Start CPU (change into Run state).
FX0-GETC does not supply any data. MS ● Green ERR ○ Off RUN ● Green	EtherCAT PLC is in Stop state.	Set EtherCAT PLC into Run state.
FX0-GETC is in Critical fault mode. MS ☼ Red (2 Hz) ERR ● Red RUN ○ Off	FX0-GETC internal device error. CPU firmware version does not support Flexi Soft gateways.	Switch off the power supply of the Flexi Soft system and switch it on again. Check the diagnostics messages with Flexi Soft Designer. Use a FX3-CPUx with the required firmware version (see section 2.2 “Correct use” on page 10). If the error remains, replace the gateway.
FX0-GETC/Flexi Soft System is in Critical fault mode MS ● Red ERR ● Red RUN ○ Off	FX0-GETC is not plugged properly into the other Flexi Soft module. Module connecting plug is soiled or damaged. Other Flexi Soft module has internal critical error.	Plug the FX0-GETC in correctly. Clean the connecting socket/plug. Repower the system. Check the other Flexi Soft modules. Check the diagnostics messages with Flexi Soft Designer.

6 Fieldbus gateways

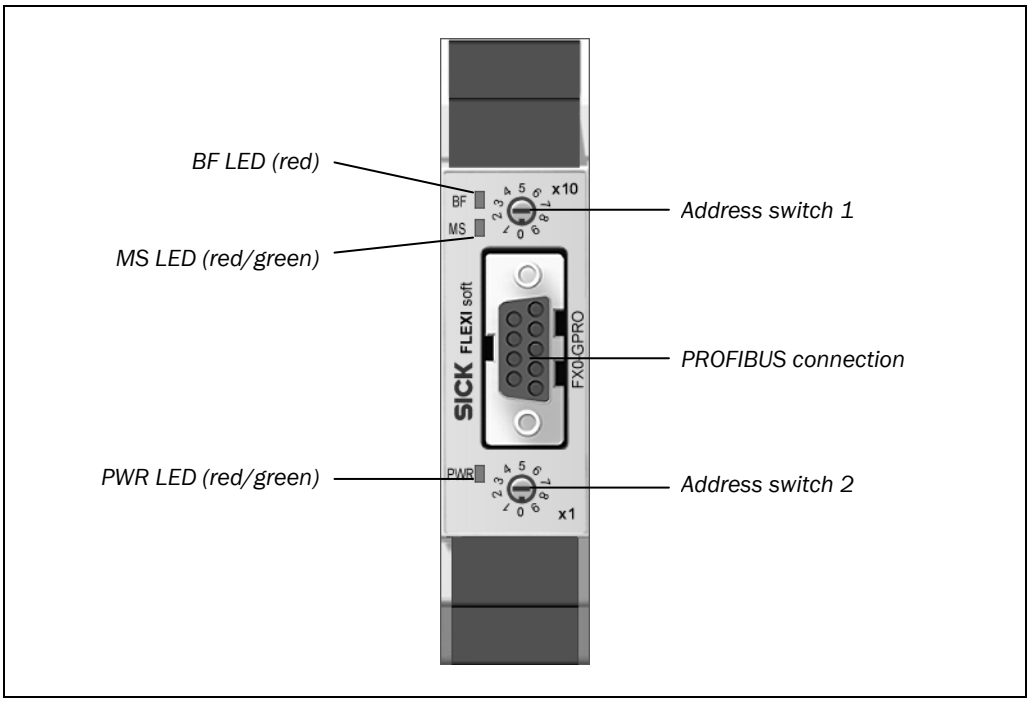
6.1 PROFIBUS DP gateway

The following Flexi Soft gateway can be used for PROFIBUS DP: FX0-GPRO.

6.1.1 Interfaces and operation

Controls and status indicators

Fig. 56: Controls and status indicators FX0-GPRO



Tab. 83: Meaning of the status LEDs of the FX0-GPRO

Symbol description:
○: LED is off
● Green: LED lights up green
☼ Red: LED flashes red

LED		Meaning
BF	○	Connection to DP master established
	● Red	No bus connection: Fieldbus cable break, address fault or master is not (or no longer) writing to the bus
MS	○	Power-up, waiting for bus off
	● Green	Executing
	☼ Green	Idle
	☼ Red/green	Executing, but there is an error at the gateway
	☼ Red	1 Hz: Configuration required or in progress 2 Hz: Critical fault on gateway
	● Red	Critical fault on another module
PWR	○	No power supply
	● Green	Power supply switched on, no error
	● Red	Critical fault

Tab. 84: Address switch FX0-GPRO

Switch	Function
× 10	Address switch 1 10 position rotary switch for setting the module address (tens)
× 1	Address switch 2 10 position rotary switch for setting the module address (ones)

How to set the PROFIBUS DP address via hardware address switches:

➢ Set the PROFIBUS DP address using the hardware address switches on the device front. Then switch the Flexi Soft system off and back on again.

How to set the PROFIBUS DP address via software using the Flexi Soft Designer:

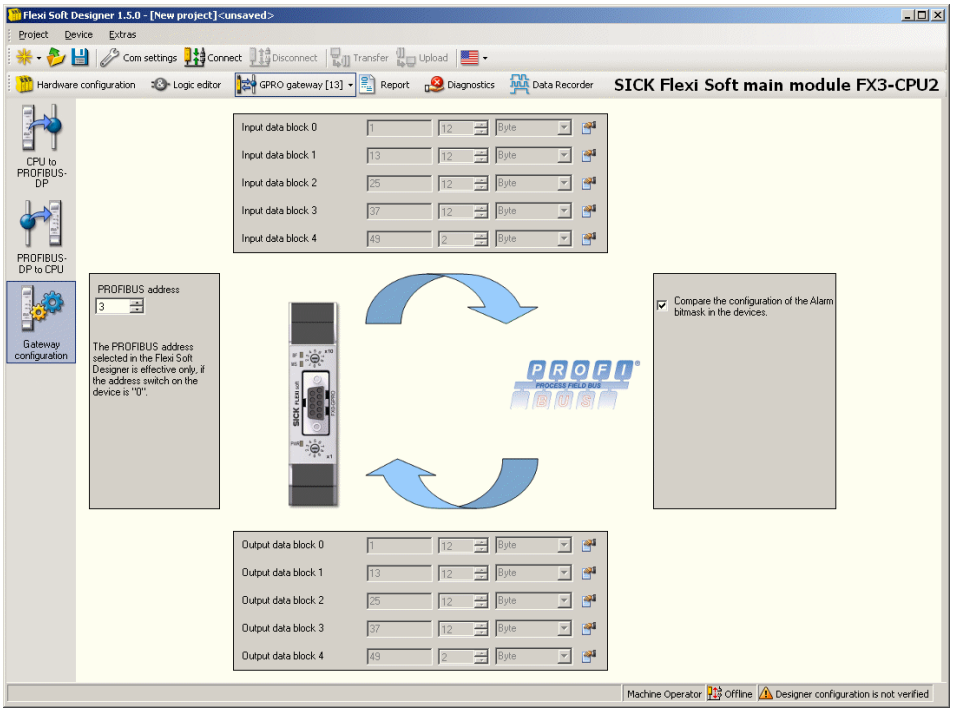
➢ Set the two hardware address switches on the device front to “00”.

➢ Open the Flexi Soft Designer and load the hardware configuration including the PROFIBUS DP gateway. Ensure your project is offline.

➢ Click on the **Gateways** button above the main window and select the FX0-GPRO or double click the FX0-GPRO in the hardware configuration view to open the gateway configuration dialog.

➢ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 57: Setting the PROFIBUS address for the FX0-GPRO



➢ Select the PROFIBUS address in the **PROFIBUS address** field.

➢ Click **Connect** to go online and transfer the configuration to the Flexi Soft system.

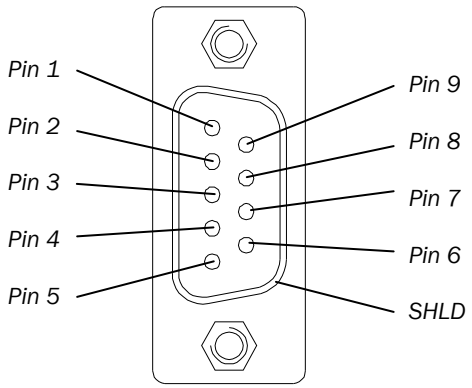
Notes

- The address that can be set via the hardware address switch ranges from 1 ... 99.
- The address that can be set via the Flexi Soft Designer software ranges from 3 ... 125.
- The PROFIBUS master cannot overwrite the address.
- A modified address setting only becomes effective after switching off and switching on the Flexi Soft system.
- In online mode, you can read the address set on the PROFIBUS DP gateway by clicking on the **Read** button above the **PROFIBUS address** field.

Plug assignment

The connection to the PROFIBUS DP fieldbus is made using a 9 pin D-Sub socket.

Fig. 58: D-Sub socket and plug pin assignments FX0-GPRO

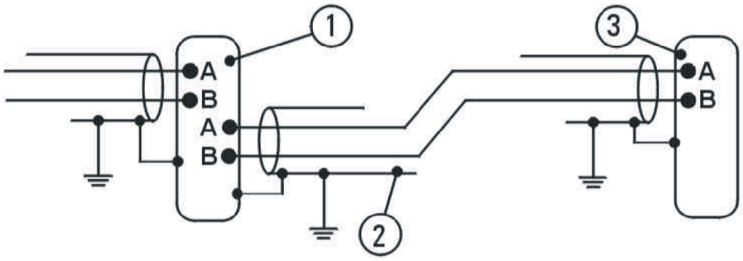


Pin	Description
1	NC
2	NC
3	RxD/TxD-P
4	CNTR-P
5	GND-EXT
6	+5V-EXT
7	NC
8	RxD/TxD-N
9	CNTR-N (GND-EXT)
SHLD	Shield

Bus cable

The bus topology for PROFIBUS DP is a linear structure comprising a screened, twisted 2 core cable with active bus termination at both ends. The possible bus lengths are 100 m at 12 Mbit/s up to 1,200 m at 94 kbit/s.

Fig. 59: Bus cable FX0-GPRO



Tab. 85: Explanation bus cable FX0-GPRO

Position	Description
1	PROFIBUS user grey
2	Shielded bus cable
3	PROFIBUS termination yellow (with integrated terminating resistors)

Cable parameters

The properties of the bus cable are specified in EN 50 170 as cable type A.

Tab. 86: Cable parameters FX0-GPRO

Property	Value
Characteristic impedance	135-165 Ω (at a frequency of 3-20 MHz)
Capacitance per unit length	< 30 pF/m
Loop resistance	≤ 110 Ω/km
Core diameter	> 0.64 mm
Core cross-section	> 0.34 mm^2

With these cable parameters, the following maximum physical sizes are possible for a bus segment:

Tab. 87: Maximum cable lengths FX0-GPRO

Baud rate [kbit/s]	Max. 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 set automatically.

The maximum baud rate is 12 Mbit/s.

Configuration comparison of the Alarm bitmask

This check box in the dialog box for the **gateway configuration** is provided for compatibility reasons and, as a rule, should not be changed. On loading configurations that have been prepared using a version of Flexi Soft Designer < 1.3.1 the check box is cleared. In the case of configurations that have been prepared using Flexi Soft Designer 1.3.1 or higher, the check box is selected.

With the check box selected the Alarm bitmasks in the device are updated by Flexi Soft Designer. This function is available for FX0-GPRO with firmware versions from V1.30.0. In this way it is possible to modify the scope of the alarms generated on the fieldbus side to new software versions on other modules.

If you change the status of this check box for a configuration already verified, then you must verify this configuration again.

6.1.2 Planning

GSD file

Normally the FX0-GPRO is used on a DP master that looks up the device characteristics in the GSD file.

You will find the GSD file and device icon for PLC interfacing with PROFIBUS support ...

- on the FX0-GPRO product page on the Internet at www.sick.com.
- in the Flexi Soft Designer program folder on your hard disk (default installation folder is “C:\programs\SICK\FlexiSoft\DeviceDescriptions\...”).

Note

There are two different GSD files available which should be used depending on the FX0-GPRO gateway’s firmware version:

Tab. 88: GSD file versions for the FX0-GPRO

FX0-GPRO firmware version	GSD file	Functionality
V1.00-V1.29	SICK0C18.gsd	DP-V0 Slave
≥ V1.30	SIC_0C18.gsd	DP-V1 Slave

FX0-GPRO gateways with firmware version ≥ V1.30 will work with the GSD file **SICK0C18.gsd** as well, but will only support DP-V0 slave functionality.

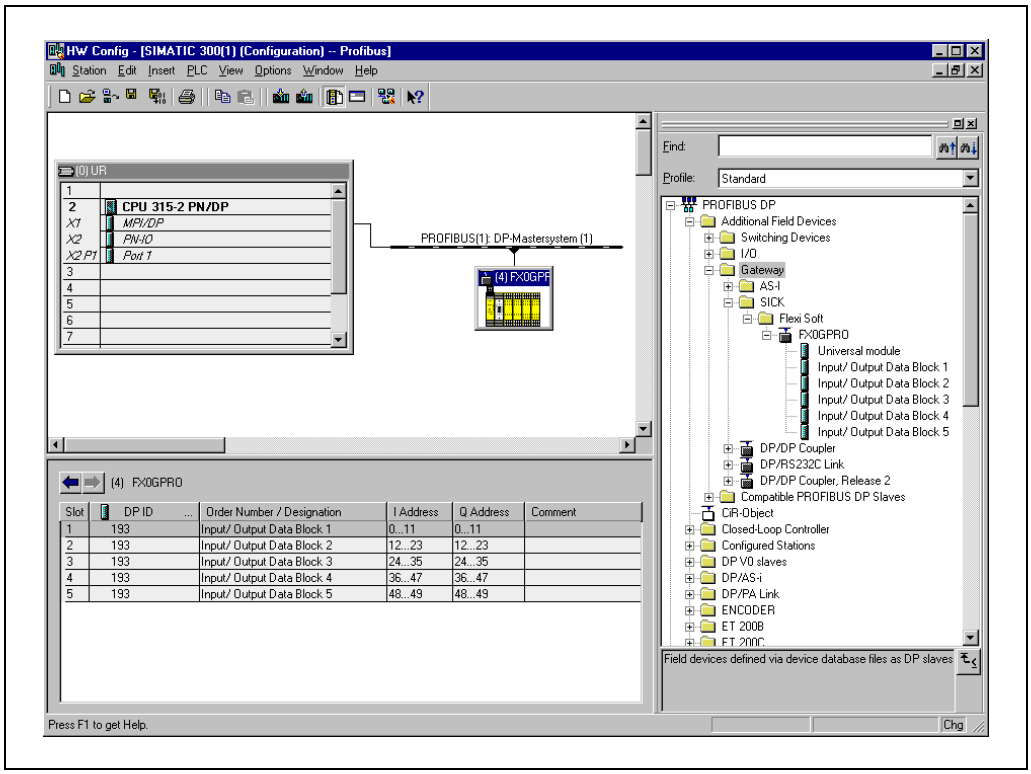
FX0-GPRO gateways with firmware versions V1.00 to V1.29 will not work with the GSD file **SIC_0C18.gsd**.

Note In DP-V1 slave mode only reading of I&M0 data is supported. This can be triggered by a Class 1 or Class 2 master.

Operational data transmitted by the FX0-GPRO PROFIBUS DP

The FX0-GPRO GSD file provides input/output data blocks (virtual I/O device modules) containing the operational data. These 5 blocks must be projected in a DP configurator in natural order (1, 2, 3, 4, 5). No other sequence is possible.

Fig. 60: PROFIBUS DP configuration example in Siemens SIMATIC Manager



Notes

- Depending on the type of PLC used, further modules may be displayed (e.g.“universal module”). These modules are not needed and should be ignored.
- The data blocks 1-4 contain 12 bytes each, data block 5 contains 2 bytes.
- The contents of the data blocks are freely selectable, but are preconfigured in the Flexi Soft Designer configuration software:

Tab. 89: Default content of input data block 1-5 of the FX0-GPRO

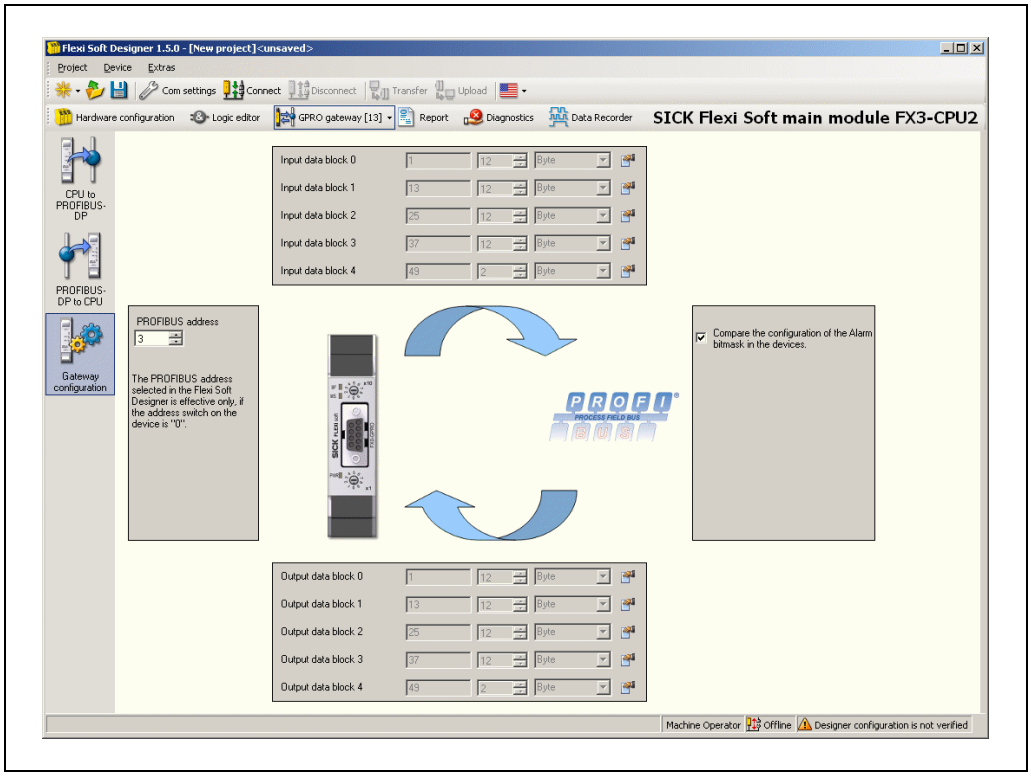
	Data block 1	Data block 2	Data block 3	Data block 4	Data block 5
	Input data	**Input data**	**Input data**	**Input data**	**Input data**
Byte 0	Input values module 1	Output values module 1	Logic result 0	Not assigned	Not assigned
Byte 1	Input values module 2	Output values module 2	Logic result 1	Not assigned	Not assigned
Byte 2	Input values module 3	Output values module 3	Logic result 2	Not assigned	Not available
Byte 3	Input values module 4	Output values module 4	Logic result 3	Not assigned	
Byte 4	Input values module 5	Output values module 5	Gateway direct output values 0	Not assigned	
Byte 5	Input values module 6	Output values module 6	Gateway direct output values 1	Not assigned	
Byte 6	Input values module 7	Output values module 7	Gateway direct output values 2	Not assigned	
Byte 7	Input values module 8	Output values module 8	Gateway direct output values 3	Not assigned	
Byte 8	Input values module 9	Output values module 9	Not assigned	Not assigned	
Byte 9	Input values module 10	Output values module 10	Not assigned	Not assigned	
Byte 10	Input values module 11	Output values module 11	Not assigned	Not assigned	
Byte 11	Input values module 12	Output values module 12	Not assigned	Not assigned	
Length	12 bytes	12 bytes	12 bytes	12 bytes	2 bytes

For detailed information about the content of the process image please see section 3.3 “Data transmitted into the network (input data sets)” on page 14.

How to set the start address for the data blocks:

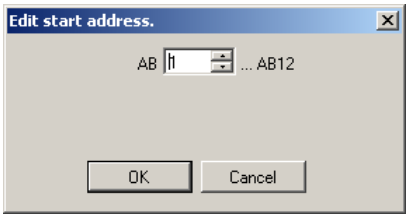
- Open the Flexi Soft Designer and load the hardware configuration including the PROFIBUS DP gateway. Ensure your project is offline.
- Click on the **Gateways** button above the main window and select the FX0-GPRO or double click the FX0-GPRO in the hardware configuration view to open the gateway configuration dialog.
- Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 61: PROFIBUS DP gateway configuration dialog



➢ Click on the button to the right of the data block for which you want to change the start address. The following dialog appears:

Fig. 62: Edit the data block start address



➢ Type in the desired new start address or use the arrows to change the address. The address set will be checked automatically for plausibility, i.e. it is not possible to configure data blocks with overlapping address ranges.

➢ Click **OK** to accept the new start address.

For further information on how to configure the process image, see chapter 7 “Layout and content of the process image” on page 173 and the Flexi Soft Designer operating instructions (SICK part no. 8012998).

6.1.3 PROFIBUS configuration of the gateway — How the data are transferred

The following steps need to be taken in order to configure the communication between PLC and gateway.

Note This document does not cover the creation of the PROFIBUS DP network or the rest of the automation system project in the network configuration tool. It is assumed the PROFIBUS project has already been set up in the configuration program, e.g. SIEMENS SIMATIC Manager. Examples refer to configurations performed with SIEMENS SIMATIC manager.

Step 1: Install the generic station description file (GSD file)

Before the FX0-GPRO can be used as device in the network configuration tool, e.g. SIEMENS SIMATIC Manager, for the first time, the generic station description (GSD) of the gateway must be installed into the hardware catalogue of the tool.

- ➢ Download the GSD file and device icon from www.sick.com, on the FX0-GPRO product page.
- ➢ Follow the instructions in the online help or in the user manual of the PROFINET network configuration tool for installing generic station description files.

Using the SIEMENS SIMATIC Manager — HW Config, the gateway then appears in the hardware catalogue under >>**PROFIBUS DP** > **Additional Field Devices** > **Gateway** > **SICK** > **Flexi Soft**.

Step 2: Add the gateway to the project

In order to have the Flexi Soft system data available in the PLC process image, the gateway must be added to the hardware configuration first. The procedure associated with this depends on the hardware configuration program of the PLC being used. On this topic, please also read the documentation for the corresponding program.

The example below shows how to add the gateway to a SIEMENS SIMATIC Manager project.

In the SIEMENS SIMATIC Hardware Manager, the gateway can be found in the hardware catalogue under >>**PROFIBUS DP** > **Additional Field Devices** > **Gateway** > **SICK** > **Flexi Soft**.

- ➢ Drag & drop the device into the PROFIBUS network. Example:

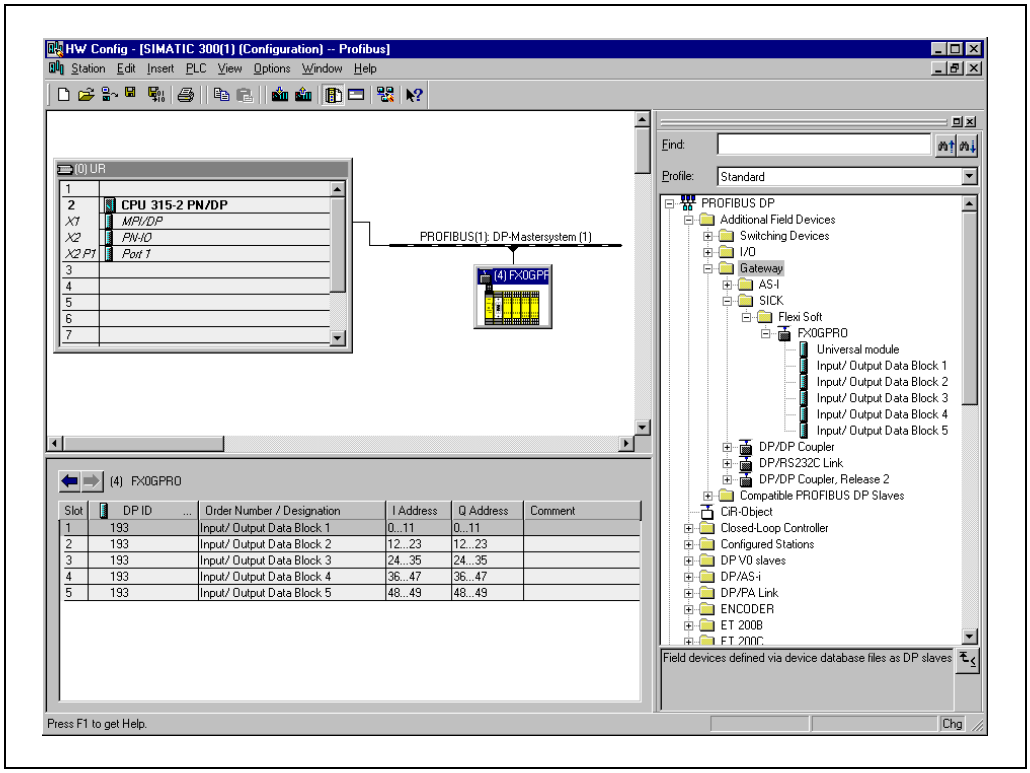


Fig. 63: PROFIBUS DP gateway in the PROFIBUS HW Config

Note The universal module does not support any data exchange. Please select only Input/ Output Data Block 1-5.

Diagnostics data FX0-GPRO PROFIBUS DP

With the FX0-GPRO, diagnostics data are available via PROFIBUS standard DP-V0 diagnostics:

- standard diagnostics (6 bytes)
- device related diagnostics: status messages or manufacturer specific messages

Each Flexi Soft module supports a unique module ID. Based on this ID the gateway determines the manufacturer specific diagnostics number. This way it is possible to retrieve module specific diagnostics texts from the GSD file. The content of the diagnostics message is shown in Tab. 90.

Tab. 90: Content of the PROFIBUS diagnostics messages

Octet	Content	Comment
7	09h	Header
8	See Tab. 91	Module number
9	0	PROFIBUS module slot number. The PROFIBUS gateway supports five slots, which do not represent the physical slots, thus all messages shall be tied to slot 0 (gateway itself).
10 (Bit 0 ... 2)	000, 001 or 010	000 = All errors going, 001 = Error coming, 010 = Error going
10 (Bit 3 ... 7)	00000...11111	Alarm sequence number, will be incremented for each state change of octet 10 Bit 0 ... 2 (error coming/going) In the case of modules with firmware V1.30 and later the alarm sequence number is not used for reasons of conformity with the PROFIBUS DP specification. These bits are therefore always 0 for these modules.
11	0 ... 14	Position of the Flexi Soft module raising the diagnostic information. 0 = FX3-CPUx 1 = 1st extension module ... 13 = 1st gateway 14 = 2nd gateway (relay modules are not counted)
12 ... 15	Variable	4 bytes module specific diagnostics data. See Tab. 92.

The following table lists the module numbers for the Flexi Soft system.

Tab. 91: Flexi Soft module numbers

Module number [dec]	Module number [hex]	Module
161	A1	Flexi Soft main module (FX3-CPUx)
162	A2	FX3-XT module (FX3-XTDI, FX3-XTIO)
163	A3	PROFIBUS gateway (FX0-GPRO)
164	A4	CANopen gateway (FX0-GCAN)
165	A5	DeviceNet gateway (FX0-GDEV)
166	A6	Modbus gateway (FX0-GMOD)
167	A7	EtherNet/IP gateway (FX0-GENT)
168	A8	PROFINET IO gateway (FX0-GPNT)
172	AC	CC-Link gateway (on request)
175	AF	Sercos III gateway (on request)
176	B0	EtherCAT gateway (FX0-GETC)
192	C0	Extension module (FX0-STIO)
193	C1	Motion control module (FX3-MOCx)
194	C2	Extension module (FX3-XTDS)

The following table matches the module specific diagnostics data (as defined by the GSD) to the appropriate error message.

The module status bits have the following significance, if not otherwise stated:

0 = Error

1 = No error

Tab. 92: PROFIBUS error messages

Module number [dec]	Diagnostics bit [Octet Bit]	Error origin	Error message
1	12.0	CPU	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration Flexi Soft system
	12.5		Power supply
	12.6		EFI1
	12.7		EFI2
	13.0		Flexi Link stations in the system 1 = All found 0 = One or more are missing
	13.1		Flexi Link stations suspended 1 = None 0 = One or more
	13.2 ... 15.7		Reserved
2	12.0	XTIO/XTDI	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Outputs power supply
	12.6		Output *Fast-Shut-Off*
	12.7		Reserved
	13.0		Input 1-2 dual channel input evaluation
	13.1		Input 3-4 dual channel input evaluation
	13.2		Input 5-6 dual channel input evaluation
	13.3		Input 7-8 dual channel input evaluation
	13.4 ... 13.7		Reserved
	14.0		External test signal Input 1
	14.1		External test signal Input 2
	14.2		External test signal Input 3
	14.3		External test signal Input 4
	14.4		External test signal Input 5
	14.5		External test signal Input 6
	14.6		External test signal Input 7
	14.7		External test signal Input 8
	15.0		Short-circuit monitoring output 1: short-circuit to high
	15.1		Short-circuit monitoring output 1: short-circuit to low
	15.2		Short-circuit monitoring output 2: short-circuit to high
	15.3		Short-circuit monitoring output 2: short-circuit to low
	15.4		Short-circuit monitoring output 3: short-circuit to high
	15.5		Short-circuit monitoring output 3: short-circuit to low
	15.6		Short-circuit monitoring output 4: short-circuit to high
	15.7		Short-circuit monitoring output 4: short-circuit to low
3	12.0	PROFIBUS gateway	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Communication from the network
	12.6		Communication to the network
	12.7 ... 15.7		Reserved

Module number [dec]	Diagnostics bit [Octet Bit]	Error origin	Error message
4	12.0	CANopen gateway	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Communication from the network
	12.6		Communication to the network
	12.7 ... 15.7		Reserved
5	12.0	DeviceNet gateway	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Communication from the network
	12.6		Communication to the network
	12.7 ... 15.7		Reserved
6	12.0	Modbus gateway	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Communication from the network
	12.6		Communication to the network
	12.7 ... 15.7		Reserved
7	12.0	EtherNet/IP gateway	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Communication from the network
	12.6		Communication to the network
	12.7 ... 15.7		Reserved
8	12.0	PROFINET gateway	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Communication from the network
	12.6		Communication to the network
	12.7 ... 15.7		Reserved
0C	12.0	CC-Link gateway	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Communication from the network
	12.6		Communication to the network
	12.7 ... 15.7		Reserved
0F	12.0	Sercos III gateway	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Communication from the network
	12.6		Communication to the network
	12.7 ... 15.7		Reserved

Module number [dec]	Diagnostics bit [Octet Bit]	Error origin	Error message
10	**12.0**	EtherCAT gateway	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Communication from the network
	12.6		Communication to the network
	12.7 ... 15.7		Reserved
20	**12.0**	STIO	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Outputs power supply
	12.6		Reserved
	12.7		Output load (overcurrent) monitoring
	13.0 ... 15.7		Reserved
21	**12.0**	MOCx	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Encoder 1
	12.6		Encoder 2
	12.7		Reserved
	13.0 ... 13.3		Reserved
	13.4		User-defined status bit 1[14)]
	13.5		User-defined status bit 2[14)]
	13.6		User-defined status bit 3[14)]
	13.7		User-defined status bit 4[14)]
	14.0 ... 15.7		Reserved

[14)] The status of this bit can be defined to suit the specific application in the MOCx logic, e.g. to indicate inadmissible movements of an axis that have been detected by an MOCx function block.

Module number [dec]	Diagnostics bit [Octet Bit]	Error origin	Error message
22	12.0	XTDS	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5		Outputs power supply
	12.6		Reserved
	12.7		Output load (overcurrent) monitoring
	13.0		Input 1-2 dual channel input evaluation
	13.1		Input 3-4 dual channel input evaluation
	13.2		Input 5-6 dual channel input evaluation
	13.3		Input 7-8 dual channel input evaluation
	13.4 ... 13.7		Reserved
	14.0		External test signal Input 1
	14.1		External test signal Input 2
	14.2		External test signal Input 3
	14.3		External test signal Input 4
	14.4		External test signal Input 5
	14.5		External test signal Input 6
	14.6		External test signal Input 7
	14.7		External test signal Input 8
	15.0 ... 15.7		Reserved
23 ... 3F	12.0	Other module	Operating state Run
	12.1		Internal tests
	12.2		Summary of bits 12.5 to 12.7
	12.3		Reserved
	12.4		Configuration of this module is valid
	12.5 ... 15.7		Reserved

6.1.4 Diagnostics and troubleshooting

For information how to perform diagnostics on the Flexi Soft system please refer to the operating instructions for the Flexi Soft Designer software (SICK part no. 8012998).

Tab. 93: Troubleshooting for the FX0-GPRO

Symbol description:
○: LED is off
● Green: LED lights up green
☼ Red: LED flashes red

Error	Possible cause	Possible remedy
The Flexi Soft Designer tool does not connect to the Flexi Soft gateway module.	FX0-GPRO has no power supply.	Establish the power supply. Check the communication settings in the Flexi Soft Designer.
FX0-GPRO does not supply any data. PWR ● Green BF ○ Off MS ☼ Red (1 Hz)	Configuration required. Configuration download is not completed.	Configure the FX0-GPRO and download the configuration to the device. Wait until the configuration download has been completed.
FX0-GPRO does not supply any data. PWR ● Green BF ○ Off MS ● Green	No data set is activated.	Activate at least one data set.
FX0-GPRO does not supply any data. PWR ● Green BF ○ Off/● Red MS ☼ Green (1 Hz)	FX0-GPRO is in Idle mode.	CPU/application is stopped. Start CPU (change into Run state)
FX0-GPRO does not supply any data. PWR ● Green BF ○ Off MS ● Green	PROFIBUS master is in stop mode.	Set PROFIBUS master into Run state.
FX0-GPRO functioned correctly after configuration, but suddenly no longer supplies data. PWR ● Green BF ● Red MS ☼ Red/green	FX0-GPRO PROFIBUS hardware address is changed. PROFIBUS cable is disconnected.	Check PROFIBUS address setting at hardware address. Check PROFIBUS cable. Check PROFIBUS master.
FX0-GPRO is in Critical fault mode. PWR ● Green BF ● Red MS ☼ Red (2 Hz)	FX0-GPRO internal device error CPU firmware version does not support Flexi Soft gateways.	Switch off the power supply of the Flexi Soft system and switch it on again. Check the diagnostics messages with Flexi Soft Designer. Use a FX3-CPUx with the required firmware version (see section 2.2 “Correct use” on page 10). If the error remains, replace the gateway.
FX0-GPRO/Flexi Soft System is in Critical fault mode PWR ● Red BF ○ Off MS ● Red	FX0-GPRO is not plugged properly into the other Flexi Soft module. Module connecting plug is soiled or damaged. Other Flexi Soft module has internal critical error.	Plug the FX0-GPRO in correctly. Clean the connecting socket/plug. Repower the system. Check the other Flexi Soft modules.

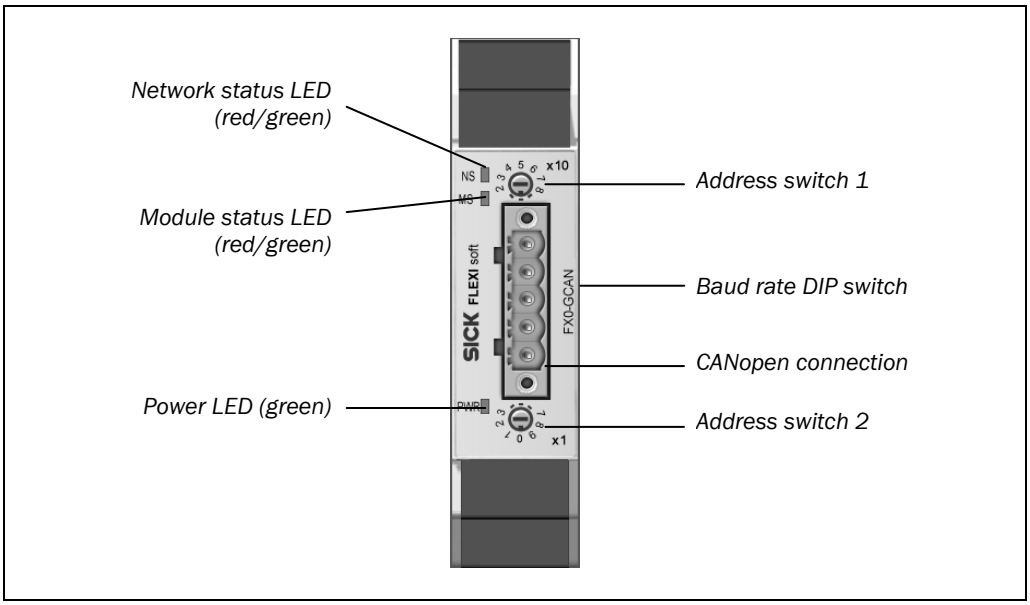
6.2 CANopen gateway

The following Flexi Soft gateway can be used for CANopen: FX0-GCAN.

6.2.1 Interfaces and operation

Controls and status indicators

Fig. 64: Controls and status indicators FX0-GCAN



Tab. 94: Meaning of the status LEDs of the FX0-GCAN

Symbol description:
○: LED is off
● Green: LED lights up green
☼● Red: LED flashes red

LED		Meaning
PWR (Power)	○	No power supply
	● Green	Operational, power supply switched on
	● Red	System error
NS (Network Status)	○	CANopen status: Stopped (except node guarding and heartbeat, if active)
	● Green	CANopen status: Operational (PDO + SDO data exchange)
	☼● Green	CANopen status: Pre-operational (SDO data exchange only)
	● Red	CAN bus off (hardware problem CAN physical layer) or Error passive
	☼● Red (1 Hz)	Node guarding failed (NMT master does not monitor slave anymore) or Heartbeat consumer failure
MS (Module Status)	○	Power-up
	● Green	Executing, FLEXBUS+ and PDO status: all “Good”
	☼● Green	Idle (cable not attached or node guarding failed)
	☼● Red/green	Executing, FLEXBUS+ and PDO status: any is “Bad”
	● Red	Critical fault, caused by emergency bit
	☼● Red (1 Hz)	Configuration required or in progress
	☼● Red (2 Hz)	Critical fault, caused by gateway itself

For diagnostics see section 6.2.14 “Diagnostics and troubleshooting” on page 159.

How to set the CANopen address via hardware address switches:

➢ Set the CANopen address using the hardware address switches on the device front. Then switch the Flexi Soft system off and back on again.

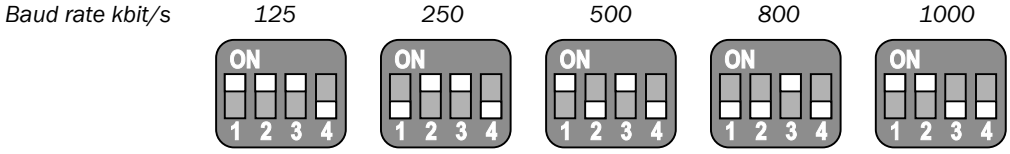
Tab. 95: Address switch FX0-GCAN

Switch	Function
× 10	Address switch 1 10 position rotary switch for setting the module address (tens)
× 1	Address switch 2 10 position rotary switch for setting the module address (ones)

How to set the baud rate via hardware DIP switches:

➢ Set the baud rate using the DIP switches on the device. Then switch the Flexi Soft system off and back on again.

Fig. 65: DIP switch settings on the FX0-GCAN



Tab. 96: DIP switch settings on the FX0-GCAN

Baud rate [kbit/s]	DIP 1	DIP 2	DIP 3	DIP 4
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

Notes

- All other DIP switch settings will set the baud rate to 125 kbit/s.
- If the *address switches* on the device are set to “00”, the DIP switch settings are ignored and the baud rate setting in the Flexi Soft Designer is used.

How to set the CANopen address and baud rate via software using the Flexi Soft Designer:

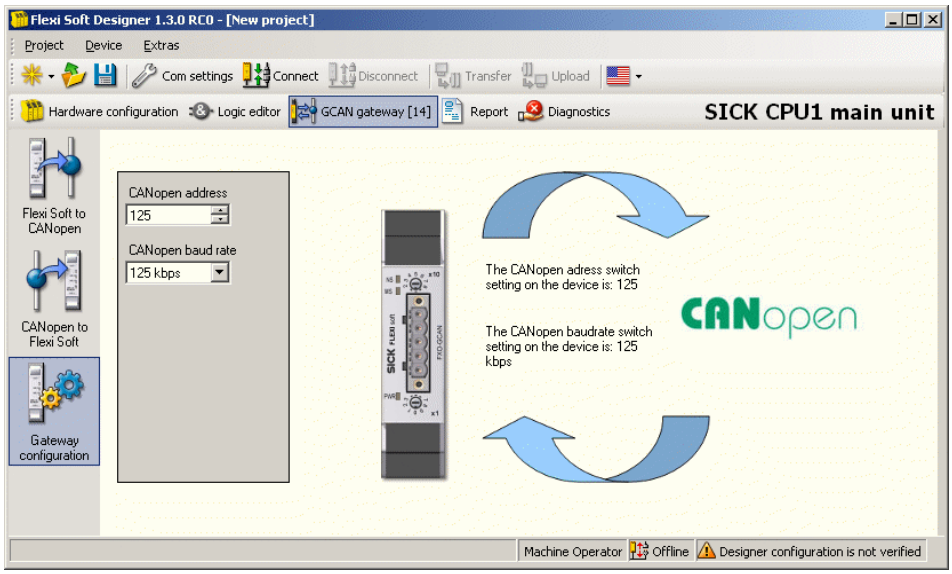
➢ Set the two hardware address switches on the device front to “00”.

➢ Open the Flexi Soft Designer and load the hardware configuration including the CANopen gateway. Ensure that your project is offline.

➢ Click on the **Gateways** button above the main window and select the FX0-GCAN or double click the FX0-GCAN in the hardware configuration view to open the gateway configuration dialog.

➢ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 66: Setting the CANopen address for the FX0-GCAN



- ➢ Select the CANopen address in the **CANopen address** field.
- ➢ Select the baud rate in the **CANopen baud rate** field.
- ➢ Click **Connect** to go online and transfer the configuration to the Flexi Soft system.

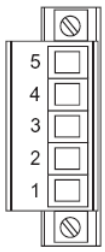
Notes

- The address that can be set via the hardware address switch ranges from 1 ... 99.
- The address that can be set via the Flexi Soft Designer software ranges from 1 ... 127.
- The CANopen master cannot overwrite the address.
- If the CANopen address and the baud rate are set using the Flexi Soft Designer, the settings will become effective immediately after transferring the configuration (i.e. without switching off and switching on the Flexi Soft system). Exception: If the system is in bus off state, a power cycle is required.

Plug assignment

The connection to the CANopen fieldbus is made using a 5 pin open style connector.

Fig. 67: Open style connector and assignment FX0-GCAN

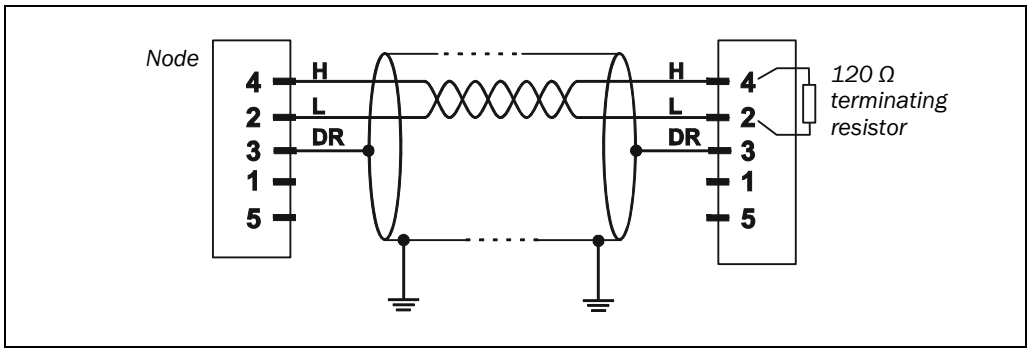


Pin	Description	
5	–	–
4	H CAN_H	CAN High
3	DR (CAN_SHLD)	Screen connection (optional)
2	L CAN_L	CAN Low
1	–	–

Bus cable

CANopen is based on a linear topology with screened, twisted pair 2 core cable and terminating resistors at both ends of the bus. The screen is connected to ground at both ends. The transmission rate, depending on the network length, is between 125 kbit/s and 1000 kbit/s. The possible network lengths are 20 m at 1000 kbit/s up to 500 m at 125 kbit/s.

Fig. 68: CANopen bus cable



Note It is not necessary to connect a voltage supply (pin 1/5) to the FX0-GCAN.

The following maximum physical sizes are possible for the network:

Tab. 97: Maximum cable lengths FX0-GCAN

Baud rate [kbit/s]	Max. cable length [m]
125	500
250	250
500	100
800	40
1000	20

EDS file

The device characteristics are described using the Electronic Data Sheet (EDS), which every standard bus configuration tool uses.

You will find the EDS file and device icon for PLC interfacing ...

- in the Internet on the FX0-GCAN product page.
- in the Flexi Soft Designer program folder on your hard disk (default installation folder is “C:\programs\SICK\FlexiSoft\DeviceDescriptions\...”).

6.2.2 CANopen configuration of the gateway — How the data are transferred

Note This document does not cover the creation of the CANopen network or the rest of the automation system project in the network configuration tool. It is assumed the CANopen project has already been set up in the configuration program, e.g. 3S Software CoDeSys 2.x. Examples refer to configurations performed with CoDeSys 2.3.

The following steps need to be taken in order to configure the communication between PLC and gateway.

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

Before the FX0-GCAN can be used as device in the network configuration tool, e.g. CoDeSys 2.3, for the first time, the electronic data sheet (EDS file) of the gateway must be installed into the hardware catalogue of the tool.

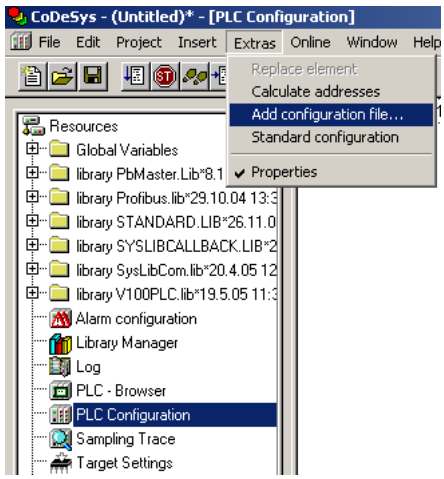
➢ Download the EDS file and device icon from www.sick.com, on the FX0-GCAN product page.

➢ Follow the instructions in the online help or in the user manual of the CANopen network configuration tool for installing EDS files.

Example — How to install the EDS file using CoDeSys 2.3:

➢ Open the **PLC Configuration** editor window.

Fig. 69: CoDeSys PLC Configuration editor window



➢ From the **Extras** menu choose the command **Add configuration file...**. A file selection dialog opens.

➢ Select the EDS file for the FX0-GCAN and click on the **Open** button.

Step 2: Add the gateway to the controller

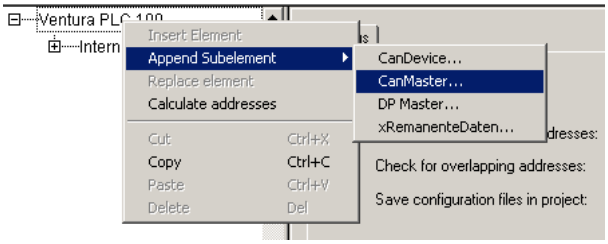
In order to have the Flexi Soft system data available in the PLC process image, the gateway must be added to the hardware configuration first. The procedure associated with this depends on the hardware configuration program of the PLC being used. On this topic, please also read the documentation for the corresponding program.

Example — How to add the FX0-GCAN using CoDeSys 2.3:

➢ Open the **PLC Configuration** editor window and select the controller.

➢ Right click on the controller to open the context menu or open the **Insert** menu.

Fig. 70: Appending a CanMaster using CoDeSys 2.3

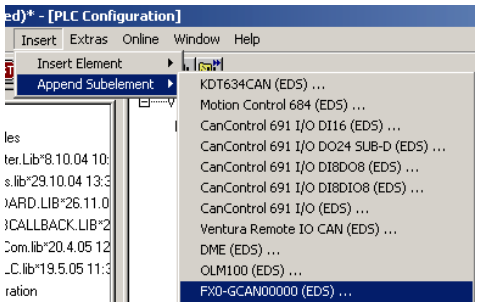


➢ In either menu, under **Append Subelement**, choose **CanMaster...**. A CanMaster is added to the controller.

➢ Now select the CanMaster.

➢ Right click on the CanMaster to open the context menu or open the **Insert** menu.

Fig. 71: Appending the FX0-GCAN using CoDeSys 2.3



➢ In either menu, under **Append Subelement**, choose **FX0-GCAN00000 (EDS)...** to append the FX0-GCAN to the CanMaster.

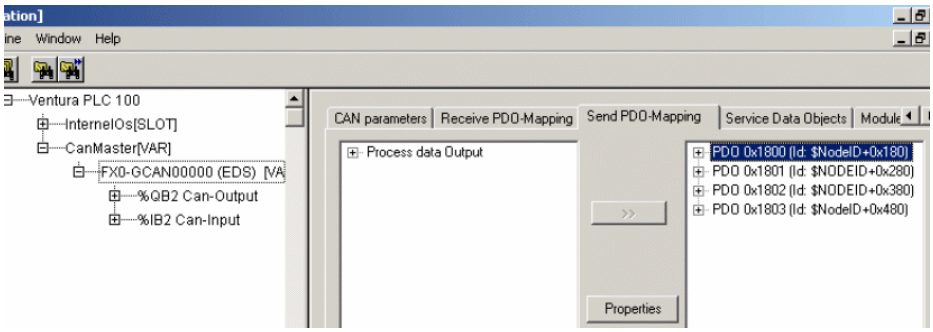
Step 3: Select and configure the process data objects (PDOs)

After adding the device to the automation network it is required to configure which of the process data objects will be used and how they will be transmitted.

Example — How to define the PDO transmission type using CoDeSys 2.3:

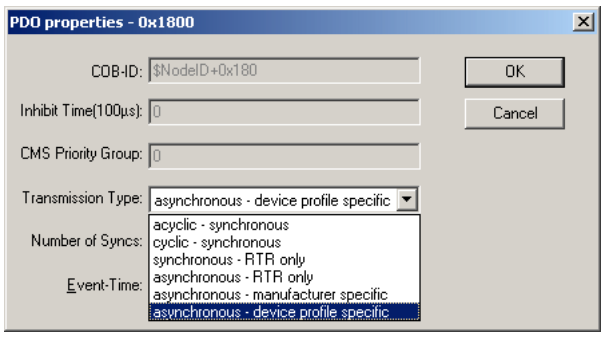
➢ In the **PLC Configuration** editor window select the FX0-GCAN. Then click on the **Send PDO-Mapping** tab on the right side.

Fig. 72: PDO configuration using CoDeSys 2.3



➢ Select one of the displayed PDOs (e.g. PDO 1) and click on the **Properties** button. The **PDO properties** dialog opens.

Fig. 73: PDO properties dialog in CoDeSys 2.3



➢ Select the desired **Transmission Type** for the PDO from the selection list, enter the **Event-Time** in ms and click **OK**. For more information please refer to section “TxPDO transmission types” on page 149 and to the manual for your CANopen system configuration software.

➢ Repeat this for the other Send-PDOs as well as for the Receive-PDOs.

6.2.3 CANopen configuration of the gateway — Which data are transferred

Each CANopen device stores its data in *objects* that are listed in the *object dictionary*. *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 as well as to control the devices. A detailed description of the different objects will be given in the following sections.

Predefined connection set (PCS)

The predefined connection set provides a simple CAN identifier structure. The FX0-GCAN gateway provides *communication objects* that can be addressed or sent using these CAN identifiers.

The PCS comprises 2 broadcast objects (NMT and SYNC) and a total of 12 peer-to-peer objects. Each of these objects has a unique 11 bit CAN identifier that consists of a function code and a device address. The device address for the broadcast objects is 0, for the other objects 1 ... 127.

Tab. 98: CAN identifier structure

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

Tab. 99: PCS communication objects

Object	CAN identifier	Meaning
Broadcast objects		
NMT	00h	Network management
SYNC	80h	Sync message
Peer-to-peer objects		
EMERGENCY	081h ... 0FFh	Status 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 object
RxSDO	601h ... 67Fh	Receive service data object
NMT-ErrorControl	701h ... 77Fh	Node guarding

Each object starts with its CAN identifier, followed by the RTR bit (Remote Transmission Request), followed by the Data Length Code (DLC), followed by 0 to 8 data bytes. The DLC (4 bits) indicates the number of data bytes.

6.2.4 NMT — Network management

The NMT broadcast object is used to start, stop or initialise CANopen devices. For this purpose one device in the CANopen network must take over the role of the NMT master. Usually this is the PLC. All other devices are regarded as NMT slaves. NMT services are broadcast services, for which the slaves do not generate a reply.

All NMT objects start with the CAN ID 00h.

Broadcast service for an NMT slave with address N:

Tab. 100: Network management for one NMT slave with address N

CAN ID	DLC	DATA							
00h	2	OP	N						

Broadcast service for all NMT slaves:

Tab. 101: Network management for all NMT slaves

CAN ID	DLC	DATA							
00h	2	OP	0						

OP	NMT operation	Explanation
80h	Change to “Pre-Operational”	After booting an NMT slave automatically enters the Pre-Operational state. In this state, communication is allowed via SDO, but not via PDO. The NMT slave can be changed from another state to this state.
01h	Change to “Operational”	The Operational state is reached from the Pre-Operational state. In this state communication via PDO is possible and the CANopen slave reacts to Sync commands. **Note:** On transition to NMT Operational state, each slave sends a TxPDO with the transmission type = 255, so that the NMT master is informed of the current input configuration.
02h	Change to “Prepared/Stopped”	Communication via SDO or PDO is not possible in this state, and there is also no reaction to SYNC commands.
81h	Change to “Reset Node”	Triggers a re-initialisation of the CANopen functionality in the NMT slave.
82h	Change to “Reset Communication”	Triggers a re-initialisation of the CANopen functionality 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) has 2 data bytes (DLC = 2). Data byte 1 contains the “Reset communication” command (82h) while data byte 2 addresses this command to all devices in the CANopen network (address = 0):

Tab. 102: Example NMT object for resetting all communication

CAN ID	DLC	DATA							
00h	2	82h	0						

6.2.5 SYNC

The SYNC command causes all TxPDOs for a CANopen slave to be sent. It is therefore possible to poll the slave using SYNC.

Tab. 103: Polling inputs using SYNC

CAN ID	DLC	DATA							
80h	0								

The slave sends all input values when this message arrives. All TxPDOs are sent.

To ensure the slave automatically sends the actual input values on the receipt of a SYNC command, the transmission type for the related PDOs must be set to 1 (cyclic, synchronous). In addition, the operating mode of the device must be “Operational”.

It is possible to change the transmission type for the TxPDOs using the SDOs 1800 ... 1803 (PDO communication parameter) and the sub-object 2. Allowed are the following types:

- acyclic/synchronous = 0
- cyclic/synchronous = 1 ... 240
- acyclic by device profile = 255 (only for TxPDO 1 ... 4, digital inputs)

6.2.6 Emergency

A CANopen slave with the address N sends an emergency message to inform the other devices about an error state.

Tab. 104: 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 low byte/high byte 7001h ... 7003h: Generic error
Err-Reg	Error register, CANopen object SDO 1001h
M1	Module number of the module causing the error (see Tab. 105)
M2 ... M5	4 bytes module specific status bits (see Tab. 105). Active bits are high (=“1”)

The following table matches the module specific diagnostics data to the appropriate error message.

Tab. 105: CANopen emergency messages

Module no. (M1)	Diagnostics bit (M2 ... M5)	Emergency origin	Emergency message
01	00	CPU	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration Flexi Soft system
	05		Power supply
	06		EFI1
	07		EFI2
	08		Flexi Link stations in the system 1 = All found 0 = One or more are missing
	09		Flexi Link stations suspended 1 = None 0 = One or more
	10 ... 31		Reserved
02	00	XTIO/XTDI	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Outputs power supply
	06		Output *Fast-Shut-Off*
	07		Reserved
	08		Input 1-2 dual channel input evaluation
	09		Input 3-4 dual channel input evaluation
	10		Input 5-6 dual channel input evaluation
	11		Input 7-8 dual channel input evaluation
	12 ... 15		Reserved
	16		External test signal Input 1
	17		External test signal Input 2
	18		External test signal Input 3
	19		External test signal Input 4
	20		External test signal Input 5
	21		External test signal Input 6
	22		External test signal Input 7
	23		External test signal Input 8
	24		Short-circuit monitoring output 1: short-circuit to high
	25		Short-circuit monitoring output 1: short-circuit to low
	26		Short-circuit monitoring output 2: short-circuit to high
	27		Short-circuit monitoring output 2: short-circuit to low
	28		Short-circuit monitoring output 3: short-circuit to high
	29		Short-circuit monitoring output 3: short-circuit to low
	30		Short-circuit monitoring output 4: short-circuit to high
	31		Short-circuit monitoring output 4: short-circuit to low
03	00	PROFIBUS gateway	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Communication from the network
	06		Communication to the network
	07 ... 31		Reserved
04	00	CANopen gateway	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Communication from the network
	06		Communication to the network

Module no. (M1)	Diagnostics bit (M2 ... M5)	Emergency origin	Emergency message
	07 ... 31		Reserved
05	**00**	DeviceNet gateway	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Communication from the network
	06		Communication to the network
	07 ... 31		Reserved
06	**00**	Modbus gateway	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Communication from the network
	06		Communication to the network
	07 ... 31		Reserved
07	**00**	EtherNet/IP gateway	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Communication from the network
	06		Communication to the network
	07 ... 31		Reserved
08	**00**	PROFINET gateway	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Communication from the network
	06		Communication to the network
	07 ... 31		Reserved
0C	**00**	CC-Link gateway	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Communication from the network
	06		Communication to the network
	07 ... 31		Reserved
0F	**00**	Sercos III gateway	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Communication from the network
	06		Communication to the network
	07 ... 31		Reserved
10	**00**	EtherCAT gateway	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Communication from the network
	06		Communication to the network
	07 ... 31		Reserved

Module no. (M1)	Diagnostics bit (M2 ... M5)	Emergency origin	Emergency message
20	**00**	STIO	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Outputs power supply
	06		Reserved
	07		Output load (overcurrent) monitoring
	08 ... 31		Reserved
21	**00**	MOCx	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Encoder 1
	06		Encoder 2
	07		Reserved
	08 ... 11		Reserved
	12		User-defined status bit 1[15)]
	13		User-defined status bit 2[15)]
	14		User-defined status bit 3[15)]
	15		User-defined status bit 4[15)]
	16 ... 31		Reserved
22	**00**	XTDS	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05		Outputs power supply
	06		Reserved
	07		Output load (overcurrent) monitoring
	08		Input 1-2 dual channel input evaluation
	09		Input 3-4 dual channel input evaluation
	10		Input 5-6 dual channel input evaluation
	11		Input 7-8 dual channel input evaluation
	12 ... 15		Reserved
	16		External test signal Input 1
	17		External test signal Input 2
	18		External test signal Input 3
	19		External test signal Input 4
	20		External test signal Input 5
	21		External test signal Input 6
	22		External test signal Input 7
	23		External test signal Input 8
	24 ... 31		Reserved
23 ... 3F	**00**	Other module	Operating state Run
	01		Internal tests
	02		Summary of bits 05 to 07
	03		Reserved
	04		Configuration of this module is valid
	05 ... 31		Reserved

Note The diagnostics bit assignment for M2 to M5 is as follows:

Tab. 106: CANopen emergency, diagnostic bits M2 to M5

Bit 0	Bit 1	...	Bit 7	Bit 8	...	Bit31
M5.0	M5.1	...	M5.7	M4.0	...	M2.7

[15)] The status of this bit can be defined to suit the specific application in the MOCx logic, e.g. to indicate inadmissible movements of an axis that have been detected by an MOCx function block.

6.2.7 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 reply within the node guarding time to the request from the NMT master. The node guarding time must be monitored by the NMT master.

The NMT master sends a CAN message with identifier <700h + node ID> and RTR bit (remote transmission request).

NMT master request:

Tab. 107: Request from NMT master

CAN ID	RTR	DLC	DATA							
700h + N	**1**	**0**								

The slave (e.g. the FX0-GCAN) then sends a status byte Byte1 with the following content:

Slave response:

Tab. 108: Response from slave

CAN ID	DLC	DATA							
700h + N	**1**	Byte1							

Tab. 109: Remote transmission request

Bit	Meaning
7	Toggle bit, changes value between two sequential requests
6 ... 0	NMT status 4 = Stopped 5 = Operational 127 = Pre-operational

Bootup

On bootup, the gateway sends a bootup message with the CAN ID 700h+N, DLC = 1 and Byte 1 = 0.

Heartbeat producer

If the gateway is configured as heartbeat producer (i.e. if SDO 1017 contains a value for the producer heartbeat time, see Tab. 119 “Supported SDOs” on page 146), it sends a cyclic message with the CAN ID 700h+N, DLC = 1 and Byte 1 = 05h. The toggle bit (Bit 7) is always 0.

Heartbeat consumer

If the gateway is configured as heartbeat consumer (i.e. if SDO 1016.1 contains a value for the consumer heartbeat time, see Tab. 119 “Supported SDOs” on page 146), at least one node guarding message must be received within the configured consumer heartbeat time (typically from an NMT master).

6.2.8 PDO communication

Process data objects (PDOs) are the real time objects of the CANopen fieldbus. They are sent without protocol overhead, i.e. no confirmation is sent from the receiver.

The FX0-GCAN provides four transmit process data objects (TxPDO) containing the operational data to be sent into the network and four receive process data objects (RxPDO) containing the operational data received from the network.

CANopen objects are addressed via 11-bit CAN identifiers. As a default, the used CAN identifier for each object derives from the object type and the configured CANopen device address. The CAN identifiers used for the PDOs can be changed using the SDOs 1400 to 1403 for the RxPDOs and SDOs 1800 to 1803 for the TxPDOs ("PDO linking").

Notes

- Each process data object contains 8 bytes.
- The contents of the process data objects are freely selectable, but are preconfigured in the Flexi Soft Designer configuration software as follows:

Tab. 110: Default content of the transmit process data objects (TxPDOs) of the FX0-GCAN

	PDO#1	PDO#2	PDO#3	PDO#4
	Input data set 1	**Input data set 2**	**Input data set 3**	**Input data set 4**
Byte 0	Logic result 0	Input values module 5	Output values module 1	Output values module 9
Byte 1	Logic result 1	Input values module 6	Output values module 2	Output values module 10
Byte 2	Logic result 2	Input values module 7	Output values module 3	Output values module 11
Byte 3	Logic result 3	Input values module 8	Output values module 4	Output values module 12
Byte 4	Input values module 1	Input values module 9	Output values module 5	Gateway direct output values 0
Byte 5	Input values module 2	Input values module 10	Output values module 6	Gateway direct output values 1
Byte 6	Input values module 3	Input values module 11	Output values module 7	Gateway direct output values 2
Byte 7	Input values module 4	Input values module 12	Output values module 8	Gateway direct output values 3

For detailed information about the content of the process image please see section 3.3 "Data transmitted into the network (input data sets)" on page 14.

For further information on how to configure the process image, see chapter 7 "Layout and content of the process image" on page 173 and the Flexi Soft Designer operating instructions (SICK part no. 8012998).

Notes

- The process data can also be written and read using service data objects SDO 6000 and SDO 6200 (see section 6.2.9 "SDO communication" on page 145). The simple SDO access is recommended for diagnostic purposes. In normal operation the faster PDO communication should be used.
- After startup or after a configuration change has been performed (either via the CANopen master or via the Flexi Soft Designer), the CANopen gateway's MS LED flashes red/green until an initial transmit/receive PDO or SDO 6000/SDO 6200 data exchange on the CANopen network has happened.

TxPDO 1...4

A transmit PDO transports data from the CANopen gateway to a CANopen device.

Tab. 111: 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 mapped into the network input data using the Flexi Soft Designer software (see 7.3 “Customizing the operational data (Flexi Soft to Network)” on page 175).

The gateway sends one or more TxPDOs if at least one of the following occurs:

- At least one input or output byte has changed its value and the transmission type for the TxPDO containing this byte has the value 255.
- At least one input or output byte has changed its value *and* the gateway receives a Sync command *and* at least one TxPDO has the transmission type 0.
- If the transmission type is n = 1...240, n Sync commands are required for the TxPDO to be sent.
- The transmission type for a TxPDO is 254 or 255 and the event timer (SDO 1800,5 for TxPDO1) has a value N > 0. In this case this TxPDO is sent every N ms.
- A TxPDO can also be polled via remote transmission request (RTR). This requires a CAN telegram to the gateway containing the CAN ID of the desired TxPDO with DLC = 0 and RTR = 1.

For all transmission methods the device operating mode must be set to “Operational” (see Tab. 101 “Network management for all NMT slaves” on page 137).

RxPDO 1...4

A receive PDO transports data from a CANopen device to the CANopen gateway.

Tab. 112: 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 mapped into the gateway input data using the Flexi Soft Designer software.

The transmission type 255 is pre-set for all RxPDOs. This means that the gateway forwards the received RxPDO data immediately to the main module. This setting cannot be changed.

6.2.9 SDO communication

SDOs are Service Data Objects. They contain a wide variety of data. These include, among other items, data on the planning or input and output data. Unlike the PDO communication, each reception of an SDO is answered at protocol level, i.e. the receiving device sends a confirmation.

In this CANopen PCS implementation the following protocols are supported:

- 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 message to server N. The 16 bit index and the sub-index for the SDO to be written are coded in this message. In addition, the request also contains 4 data bytes with the data to be written.

Tab. 113: 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 sub-index

The server then replies with a confirmation message:

Tab. 114: Write SDO 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 message contain zeros.

SDO upload expedited (Read SDO)

The client requests the contents of an SDO with a request message to server N. The 16 bit index and the sub-index for the SDO to be read are coded in this message. Byte 1 to 4 in the read request message contain zeros.

Tab. 115: 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 will reply with the following message. Byte 1 to 4 contain the value of the requested object.

Tab. 116: Read SDO confirmation

CAN ID	DLC	Data							
580h + N	**8**	43h	SDO_L	SDO_H	SUB	Byte 1	Byte 2	Byte 3	Byte 4

CANopen data types UDINT and UINT

In order to transfer data types UDINT or UINT, the data must be in Intel format. E.g. the 32 bit value 12345678h must be transferred in the data bytes 5, 6, 7 and 8 in the following order: [5] = 78, [6] = 56, [7] = 34, [8] = 12.

Note This applies also to the SDO index in data byte 2 and 3 which has the data type UINT. I.e. the low byte is transferred in data byte 2 and the high byte is transferred in data byte 3.

Example: The following messages are necessary to read the 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:

Tab. 117: Example — Read SDO

CAN ID	DLC	Data							
602h	**8**	40h	03h	10h	01h	00h	00h	00h	00h

The server responds:

Tab. 118: Example — Read SDO confirmation

CAN ID	DLC	Data							
582h	**8**	43h	03h	10h	01h	08h	00h	50h	02h

The response data combine to the 32 bit word 02500008h.

6.2.10 SDO object directory

Every CANopen device manages its SDOs in an object directory. The complete object directory is formally described in an EDS file. Many CANopen tools can read this EDS file and as a result know the object characteristics of the CANopen device.

In the following table, all SDOs for the FX0-GCAN gateway are shown.

Tab. 119: 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
1014	COB ID EMCY (available version V1.30.0 or higher)
1016	Consumer heartbeat time
1017	Producer heartbeat time
1018	Identity record
1027	Module list
1400...1403	RxPDO 1...4 communication parameters
1600...1603	RxPDO 1...4 mapping parameters
1800...1803	TxPDO 1...4 communication parameters
1A00...1A03	TxPDO 1...4 mapping parameters
3100	Module status bits
3200	Config CRC array
3300	Module type code array
6000	Process data input objects
6200	Process data output objects

You can find more detailed information on these SDOs in the CANopen draft standard DS 301 V4.02 (DSP 301 V4.1).

SDO 1001: Error register

The error register (SINT) contains an error bit indicating whether an error is present. If bit 0 is set to 1, a “generic error” has been detected.

SDO 1003: Error list (error history)

SDO 1003 is an array containing the last 10 error codes that have been reported by the gateway via emergency message. Array index 0 contains the number of error codes that have been recorded in SDO 1003.

A new error will be recorded in index 1, while older errors will be renumbered (increased by 1). The array index can be overwritten externally with a 0, which will clear the array completely.

Notes

- Not all errors that are reported via emergency message will be recorded in SDO 1003, but only the errors listed in Tab. 105.
- Entries in SDO 1003 are UDINT and normally divided in 16 bit error code and 16 bit additional information. In case of an emergency, the module status diagnostics (4 byte) are entered here.

SDO 1005: COB ID SYNC

SDO 1005 contains the COB ID of the Sync object. As a default this value is 80h, but it can be changed.

Note When you change the COB ID of the Sync object, keep in mind that the new ID must not be assigned to another communication object already.

SDO 1008: Device name

SD0 1008 contains a device name (VISIBLE STRING).

Note This SDO can not be read using a simple “SDO upload expedited”. Instead, the “Upload SDO segment protocol” (client command specifier ccs = 3) must be used as described in the CANopen specification DS 301.

SDO 1009: Hardware version

SDO 1009 contains the current hardware version of the device (VISIBLE STRING).

Note This SDO can not be read using a simple “SDO upload expedited”. Instead, the “Upload SDO segment protocol” (client command specifier ccs = 3) must be used as described in the CANopen specification DS 301.

SDO 100A: Software version

SDO 100A contains the current software version of the device (VISIBLE STRING).

Note This SDO can not be read using a simple “SDO upload expedited”. Instead, the “Upload SDO segment protocol” (client command specifier ccs = 3) must be used as described in the CANopen specification DS 301.

SDO 100C: Guard time

The product of guard time (UINT) and life time factor (SINT) results in the life guarding time.
Life guarding time (ms) = guard time (ms) × life time factor
The master must send a node guarding message to the slave at least once during the life guarding time. If the life guarding time is exceeded (life-guarding error), the gateway generates a cable break error and sets all process data coming from the network to 0; the NS LED will start flashing Red.

Life guarding is activated in the slave by the first node guarding message if the set life guarding time is not 0. If after activation of the life guarding the guard time or the life time factor is set to 0, life guarding is deactivated. See also section 6.2.11 “Guarding protocols” on page 153.

SDO 100D: Life time factor

SDO 100D contains the life time factor (SINT). See SDO 100C.

Note The life time factor must be either = 0 (disabled) or ≥ 1.5.

SDO 1016: Consumer heartbeat time

The gateway is configured as a heartbeat consumer if SDO 1016 contains a value higher than 0 for the consumer heartbeat time. The consumer heartbeat time is defined in ms.

The NMT master must send at least one node guarding message to the slave within this time. If the consumer heartbeat time is exceeded (life-guarding error), the gateway generates a cable break error and sets all process data coming from the network to 0; the NS LED will start flashing Red.

SDO 1017: Producer heartbeat time

The gateway can also function 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 working correctly.

The producer heartbeat time is defined in ms. For internal processing it is rounded 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.

Note It is not possible to use heartbeat signals and life guarding messages at the same time because both functions use the same CAN identifier.

See also section 6.2.11 “Guarding protocols” on page 153.

SDO 1018: Identity record

This SDO contains basic information about the gateway.

Tab. 120: SDO 1018 contents

Subindex	Mapping	Format	Description
1	Vendor ID	UDINT	Unique identification of the manufacturer (e.g. SICK)
2	Product code	UDINT	Device variant
3	Revision number	UDINT	Software version of the device
4	Serial number	UDINT	Serial number of the device

SDO 1027: Module list

The module list contains the module type and the module ID of all Flexi Soft modules in the system.

Tab. 121: SDO 1027 contents

Subindex	Module	Format
1	Main module (FX3-CPU0, -CPU1 or -CPU2)	SINT
2...13	Extension modules (XTIO, XTDI or MOCx)	SINT
14, 15	Gateways	SINT

You can find the module types and module IDs in Tab. 105. The return value for free module slots is 0.

SDO 1400...1403: RxPDO communication parameters

Using SDO 1400 to 1403 the communication parameters for the RxPDOs 1 to 4 are configured. E.g. SD0 1400 defines the parameters for RxPDO 1 etc.

Tab. 122: SDO 1400...1403 contents

Subindex	Mapping	Format	Description
1	COB ID	UDINT	CAN identifier for this PDO, read-only
2	Receive mode	SINT	Fix 255 (asynchronous mode)

The receive mode (read-write) defines how the PDO shall be received. For the RxPDOs the receive mode is set to 255 (asynchronous mode). In this mode the data of a received RxPDO will be routed immediately to the outputs.

Note If the receive mode is set to another value than 255, an error code is generated (abort code 0609 0030h, invalid parameter value).

SDO 1600...1603: RxPDO mapping parameters

This SDO can not be used since the RxPDO mapping is done using the Flexi Soft Designer. See also Tab. 110 and Tab. 112.

SDO 1800...1803: TxPDO communication parameters

Using SDO 1800 to 1803 the communication parameters for the TxPDOs 1 to 4 are configured. E.g. SD0 1800 defines the parameters for TxPDO 1 etc.

Tab. 123: SDO 1800...1803 contents

Subindex	Mapping	Format	Description
1	COB ID	UDINT	CAN identifier for this PDO, read-only
2	Transmission type	SINT	Defines when the PDO is to be sent
5	Event timer	UINT	In ms

The transmission type is set to 255 (asynchronous mode, event triggered) as a default for all TxPDOs.

The event timer contains the rate in ms for the cyclic transmission of the TxPDO.

TxPDO transmission types

Tab. 124: TxPDO transmission types

TxPDO	Synchronous	Asynchronous	RTR
1, 2, 3, 4	0, 1...240	254, 255	253

Note If the transmission type is set to an invalid value, an error code is generated (abort code 0609 0030h, invalid parameter value).

Synchronous: The synchronous transmission type 0 means that the TxPDO will be sent after the reception of a Sync message, but only if data has changed. The synchronous transmission types n = 1...240 define that the TxPDO will be sent after the reception of the n^{th} Sync message.

Asynchronous, event triggered on change of state: The asynchronous transmission type 255 (without configured event timer) means that the TxPDO will be sent each time if at least one input bit has changed that has been mapped on this PDO.

Asynchronous, event triggered on timer event: The asynchronous transmission type 254/255 (with configured event timer) defines that the TxPDO will be sent each time the event timer has expired. E.g. a value of 500 for the event timer means that the gateway will send the corresponding TxPDO each 500 ms.

RTR, on request: The transmission type 253 defines that the TxPDO can be polled via RTR (remote transmission request). This requires a CAN message with DLC = 0, RTR = 1 and the COB ID of the TxPDO to the gateway. The gateway will then respond with the requested TxPDO.

SDO 1A00...1A03: TxPDO mapping parameters

This SDO can not be used since the TxPDO mapping is done using the Flexi Soft Designer. See also Tab. 110 and Tab. 111.

SDO 3100: Module status bits

SDO 3100 contains the module status bits from the Flexi Soft system (see Tab. 105). Active bits are low (=“0”).

Tab. 125: SDO 3100 contents

SDO array	Data set parameter	Module	Size
3100,1	Module status module 0	CPU	UDINT
3100,2	Module status module 1	Extension	UDINT
...	...	...	...
3100,14	Module status module 13	Gateway	UDINT
3100,15	Module status module 14	Gateway	UDINT

Note The module positions in the Flexi Soft Designer are numbered from 0 to 14. Therefore the subindex for SDO 3100 is = module position + 1.

SDO 3100 is read-only.

SDO 3200: Config CRC array

SDO 3200 contains the system CRCs in UDINT format.

SDO 3300: Module type code array

SDO 3300 contains the type codes of the max. 15 modules in the Flexi Soft system in SINT format (8 bytes per module = 120 bytes). See the following table for details.

Tab. 126: Module type codes in SDO 3300

Byte	Bit	Value	Short	Designation
0	**0-3**	**System**		
		07h	FX	Flexi Soft controller
	4-7	**Safety category**		
		00h	0	None
		01h	1	SIL1
		02h	2	SIL2
		03h	3	SIL3
1	**0-7**	**Module type**		
		00h	CPU0	Main module of the Flexi Soft safety controller
		01h	CPU1	Main module of the Flexi Soft safety controller with EFI
		02h	CPU2	Main module of the Flexi Soft safety controller with EFI and ACR
		03h	CPU3	Main module of the Flexi Soft safety controller with EFI, ACR and Flexi Link
		04h	XTDI	Input extension module
		05h	XTDS	Input/output extension module with safe inputs and non-safe outputs
		06h	XTIO	Input/output extension module
		07h	GPRO	PROFIBUS DP gateway
		08h	GDEV	DeviceNet gateway
		09h	GCAN	CANopen gateway
		0Ah	GENT	EtherNet/IP gateway
		0Bh	GMOD	Modbus TCP gateway
		0Ch	GPNT	PROFINET IO gateway
		14h	GCC1	CC-Link gateway
		15h	GS3S	Sercos III gateway
		16h	GETC	EtherCAT gateway
		20h	STIO	Input/output extension module with non-safe inputs and non-safe outputs
		21h	MOC1	Motion control module with position monitoring
		24h	MOC0	Motion control module
		FFh	Empty	No module type (empty configuration)
2-6	**0-7**	**Internal use**		

Byte	Bit	Value	Short	Designation
7	0-7	**Module identification for diagnostic purposes**		
		00h	–	–
		01h	FX3-CPUxxxxxx	Main module of the Flexi Soft safety controller
		02h	FX3-XTDIxxxxx	Flexi Soft extension module with safe inputs
			FX3-XTIOxxxxx	Flexi Soft extension module with safe inputs and safety outputs
		03h	FX0-GPROxxxxx	Flexi Soft PROFIBUS DP gateway
		04h	FX0-GCANxxxxx	Flexi Soft CANopen gateway
		05h	FX0-GDEVxxxxx	Flexi Soft DeviceNet gateway
		06h	FX0-GMODxxxxx	Flexi Soft Modbus TCP gateway
		07h	FX0-GENTxxxxx	Flexi Soft EtherNet/IP gateway
		08h	FX0-GPNTxxxxx	Flexi Soft PROFINET IO gateway
		0Ch	FX0-GCC1xxxxx	Flexi Soft CC-Link standard gateway
		0Fh	FX3-GS3Sxxxxx	Flexi Soft Sercos III gateway
		10h	FX0-GETCxxxxx	Flexi Soft EtherCAT standard gateway
		20h	FX0-STIOxxxxx	Flexi Soft extension module with non-safe inputs and non-safe outputs
		21h	FX3-MOCxxxxx	Flexi Soft motion control module
		22h	FX3-XTDSxxxxx	Flexi Soft extension module with safe inputs and non-safe outputs

SDO 6000: Process data input objects

The 32 byte process input data can be written in SDO array 6000. These are the same data as in RxPDO 1-4 (see section 6.2.8 “PDO communication” on page 143). The mapping is as follows:

Tab. 127: 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 is write-only.

SDO 6200: Process data output objects

The 32 byte process output data can be read from SDO array 6200. These are the same data as in TxPDO 1-4 (see section 6.2.8 “PDO communication” on page 143). The mapping is as follows:

Tab. 128: 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 is read-only.

6.2.11 Guarding protocols

CANopen provides several possibilities for actively monitoring the correct function of the fieldbus interface (e.g. cable break detection).

WARNING

Always use either node guarding or heartbeat!

According to the CIA CANopen specification DS 301 guarding is mandatory. Please activate either node guarding or heartbeat. If no guarding is configured the Flexi Soft system can not detect an interruption of the CANopen communication, e.g. a broken network cable. In this case the CANopen gateway’s input and output data may be frozen.

Heartbeat

A heartbeat producer is a CANopen device that sends a cyclic heartbeat message. This enables all other CANopen devices to detect whether the heartbeat producer is still working correctly and what is its current status. Heartbeat messages are sent in a regular time interval, the producer heartbeat time, which can be configured using SDO 1017. The configured 16 bit value will be rounded to the next multiple of 4 ms.

A heartbeat consumer is a CANopen device that expects a cyclic node guarding message within a certain time interval, the consumer heartbeat time, which can be configured using SDO 1016. If the heartbeat consumer receives no node guarding message within the configured consumer heartbeat time, it sends a life guarding emergency message and sets the incoming process data to 0. Additionally the gateway generates an internal “cable break” error that can be processed by the main module.

Node guarding

Node guarding is performed by an NMT master. This can be each CANopen device that is able to perform this function as a client. The NMT master sends a cyclic node guarding message to the monitored device which must respond within a defined period that is monitored by the NMT master. If the monitored device does not answer within the node life time, the NMT master regards this as a malfunction of the device and takes the appropriate measures.

Life guarding

Life guarding is performed by the gateway itself. The node life time is calculated in the gateway from the values of SDO 100C (guard time) and SDO 100D (life time factor). If the gateway does not receive a node guarding message at least once within this node life time from an NMT master, the gateway generates an internal “cable break” error that can be processed by the CPU main module and the NS LED will start flashing Red.

Notes

- Cable break detection is possible for the gateway either if life guarding is activated, i.e. if both objects SDO 100C and SDO 100D have a value other than 0. In this case life guarding starts when the first node guarding request is received from an NMT master and ends if the master sends a “Reset communication”.
- Alternatively, cable break detection is possible if the gateway is configured as a heartbeat consumer. In this case, the cable break detection is performed by the gateway itself.
- Heartbeat (producer) works without node guarding. In this case, the gateway can not detect a fieldbus cable break.
- Heartbeat and node guarding/life guarding can not be used simultaneously.
- If the configuration is changed from activated life guarding to no life guarding or vice versa a complete power reset of the Flexi Soft system is required to setup the CANopen network communication properly.

The following table gives 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. 129: Guarding protocol overview and comparison

SDO 1016	SDO 1017	SDO 100C × 100D	Gateway heartbeat	Gateway life guarding	NMT master node guarding
0	**0**	**0**	Not allowed: Always use 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 heartbeat consumer is possible
> 0	**> 0**	**0**	**Cyclic heartbeat (producer and consumer)**	Cable break detection	Not possible
> 0	**> 0**	**> 0**	**Not allowed**		

Note It is not useful to use heartbeat and life guarding at the same time.

6.2.12 Error objects

The FX0-GCAN reports CAN specific errors (e.g. initialisation errors, cable break, CAN communication errors) as FLEXBUS+ errors to the main module.

Module specific errors as described in Tab. 105 are reported as extended diagnostics via the emergency object and SDO 1003.

Emergency object

The emergency producer (CANopen gateway) is triggered to send the emergency object to the emergency consumer (any CANopen device, normally the controller) by the occurence of CAN specific errors or an error condition as described in Tab. 105.

The emergency object is being sent as described in DS 301 (section 9.2.5):

Tab. 130: Emergency states and transitions

Emergency state before	Transition	Module specific alarms	Emergency state after
Error free	1	Incoming error	Error occured
Error occured	2	Outgoing error, other errors present	Error occured
Error occured	3	Incoming error, other errors present	Error occured
Error occured	4	All errors cleared	Error free

The gateway is in one of two emergency states, either *Error free* or *Error occured*. Emergency objects are sent depending on the transitions between these two emergency states. The error code in the emergency object shows the emergency state the gateway is in. See also Tab. 131.

Error objects overview

CAN specific errors	Error code FLEXBUS+	Error type	Emergency error code Error register M1...M5	Error history SDO 1003	Result/possible remedy
CAN data overrun CAN controller overrun in Rx Fifo	4501h	Warning	8110h 11h 1, 0, 0, 0, 0	–	CAN messages have been lost. Bandwidth is limited. Check CAN settings, increase baud rate, reduce participants/traffic.
CAN error passive CAN controller is in error passive state.	4503h	Warning	8120h 11h 0, 0, 0, 0, 0	–	The gateway sends only recessive bits, i.e. it sets its own messages to invalid. The reason is either a gateway hardware error or external data transmission interferences. Check cabling.
CAN bus off CAN controller is in bus off state	4504h	Warning	–	–	Massive transmission errors. CAN controller has disconnected from the bus. Possible hardware defect. Power cycle the Flexi Soft system.
CAN Tx Fifo overflow CAN controller has no transmit ressources	4506h	Warning	8110h 11h 2, 0, 0, 0, 0	–	CAN messages that were to be sent by the gateway have been lost. The number of events triggering the gateway to send CAN messages is too high for the set baud rate. Increase the baud rate or change the gateway configuration.
CAN init failed The CAN controller could not be initialised	C507h	Critical	–	–	The CAN controller/transceiver is possibly defect. Replace the FX0-GCAN with a new device.
CANopen life guarding CANopen life guarding has detected a cable break	4508h	Warning	8130h 11h 0, 0, 0, 0, 0	–	Life guarding error message generated by the gateway: Either a failure of the node guarding or heartbeat NMT master has occurred or the CAN cable is broken. Check the CANopen master. Check cabling.

Module specific alarms	Error code FLEXBUS+	Transition emergency state chart	Emergency error code Error register M1...M5	Error history SDO 1003	
Gateway detected incoming error as per trigger conditions	–	1	FF01h 81h M1 = Module index M2...M5 = Module diagnostics data	M2, M3, M4, M5	See Tab. 105.
Gateway detected outgoing error, other errors are present	–	2	FF02h 81h M1 = Module index M2...M5 = Module diagnostics data	M2, M3, M4, M5	See Tab. 105.
Gateway detected incoming error, other errors are present	–	3	FF03h 81h M1 = Module index M2...M5 = Module diagnostics data	M2, M3, M4, M5	See Tab. 105.
All errors cleared	–	4	0000h 00h M1 = 0 M2...M5 = 0	–	

Tab. 131: Error objects overview

6.2.13 CANopen diagnostics examples

Example 1: XTIO module on position 1, output Q4 stuck at high

The gateway sends an emergency message (see Tab. 104).

CAN ID	DLC	DATA							
08C	**8**	03	FF	01	01	40	00	00	00

The gateway's CANopen address is 12 (= C hex). The XTIO module has the position 1 in the Flexi Soft system.

08C: Identifier (80 + C)

8: Data length code: 8 bytes follow.

03FF: Error code FF03: device specific error

01: Error register 01 of SDO 1001H

01: Module index M1: module in position 1

40: Module status bit 30 (bit 6 of byte M2) = 1: output 4 stuck at high (see Tab. 105)

Reading the current error from SDO 3100:

PLC request:

CAN ID	DLC	DATA							
60C	**8**	40	00	31	02	00	00	00	00

60C: Identifier (600 + C)

8: Data length code: 8 bytes follow.

40: Expedited upload request

00 31: Index 3100

02: Subindex 02: module in position 1 (see Tab. 125)

Gateway response:

CAN ID	DLC	DATA							
58C	8	42	00	31	02	BF	FF	FF	FB

58C: Identifier (580 + C)

8: Data length code: 8 bytes follow.

42: Upload response, data set size is not indicated

00 31: Index 3100

02: Subindex 02: module in position 1 (see Tab. 125)

FB: Error byte M5, bit 2 = 0: external error

BF: Error byte M2, bit 30 = 0: output 4 stuck-at-high error

Reading the error from the error history in SDO 1003:

PLC request:

CAN ID	DLC	DATA							
60C	8	40	03	10	01	00	00	00	00

60C: Identifier (600 + C)

8: Data length code: 8 bytes follow.

40: Expedited upload request

03 10: Index 1003

01: Subindex 01: 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: 8 bytes follow.

42: Upload response, data set size is not indicated

03 10: Index 1003

01: Subindex 01: last error

40: Module status bit 30 (bit 6 of byte M2 = 0: output 4 stuck at high

Example 2: XTDI module dual channel input error on I1/I2

The gateway sends an emergency message (see Tab. 104).

CAN ID	DLC	DATA							
08C	8	03	FF	01	0C	00	00	01	00

The gateway's CANopen address is 12 (= C hex). The XTDI module has position 12 in the Flexi Soft system.

08C: Identifier (80 + C)

8: Data length code: 8 bytes follow.

03FF: Error code FF03: device specific error

01: Error register 01 of SDO 1001H

0C: Module index M1: module in position 12 (C hex)

01: Module status bit 8 (bit 0 of byte M4) = 1: input 1-2 dual channel input evaluation: error detected (see Tab. 105)

Reading the current error using SDO 3100:

PLC request:

CAN ID	DLC	DATA							
60C	8	40	00	31	0D	00	00	00	00

60C: Identifier (600 + C)

8: Data length code: 8 bytes follow.

40: Expedited upload request

00 31: Index 3100

0D: Subindex 0D = module in position 12 (module position = subindex – 1, see also Tab. 125)

Gateway response:

CAN ID	DLC	DATA							
58C	8	42	00	31	0D	FF	FF	FE	FB

58C: Identifier (580 + C)

8: Data length code: 8 bytes follow.

42: Upload response, data set size is not indicated

00 31: Index 3100

0D: Subindex 0D: module in position 12 (see Tab. 125)

FB: Error byte M5, bit 2 = 0: external error

FE: Error byte M4, bit 0 = 0: input 1-2 dual channel input evaluation: error detected (see Tab. 105)

Reading the error from the error history in SDO 1003:

PLC request:

CAN ID	DLC	DATA							
60C	8	40	03	10	01	00	00	00	00

60C: Identifier (600 + C)

8: Data length code: 8 bytes follow.

40: Expedited upload request

03 10: Index 1003

01: Subindex 01: 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: 8 bytes follow.

42: Upload response, data set size is not indicated

03 10: Index 1003

01: Subindex 01: last error

01: Module status bit 8 (bit 0 of byte M4) = 0: input 1-2 dual channel input evaluation: error detected

6.2.14 Diagnostics and troubleshooting

For information on how to perform diagnostics on the Flexi Soft system please refer to the operating instructions for the Flexi Soft Designer software (SICK part no. 8012998).

Tab. 132: Troubleshooting for the FX0-GCAN

Symbol description:
○: LED is off
● Green: LED lights up green
☼ Red: LED flashes red

Error	Possible cause	Possible remedy
FX0-GCAN does not supply any data. PWR ● Green NS ○ Off MS ☼ Red (1 Hz)	Configuration required, node guard or heartbeat message have not been sent. Configuration download is not completed.	Configure the FX0-GCAN and download the configuration to the device. Wait until the configuration download has been completed.
FX0-GCAN does not supply any data. PWR ● Green NS ● Green MS ☼ Red (1 Hz)	Configuration download is not completed.	Wait until the configuration download has been completed.
FX0-GCAN does not supply any data. PWR ● Green NS ● Green MS ☼ Red/green	No PDO transfer since power-up.	Start PDO transfer. Transfer PDO via SDO 6000 or 6200.
FX0-GCAN does not supply any data. PWR ● Green NS ☼ Green MS ☼ Red/green	No PDO Transfer since power-up. False baud rate (CAN transceiver could be in passive error). False Node ID/CANopen address. CAN cable has been disconnected.	Start PDO transfer. Transfer PDO via SDO 6000 or 6200. Check and correct the baud rate. Check and correct the address. Check CANopen cabling.
FX0-GCAN does not supply PDO data. PWR ● Green NS ○ Off/● Red/● Green MS ☼ Green (1 Hz)	FX0-GCAN is in Idle mode. Node guard or heartbeat messages will be sent. Flexi Soft configuration is not verified and CPU module is stopped.	CPU/application is stopped. Start CPU (change into Run state). Verify the configuration with Flexi Soft Designer and start the CPU module.
FX0-GCAN does not supply PDO data. PWR ● Green NS ● Green MS ○ Off	Supply voltage too low.	Check supply voltage.

Error	Possible cause	Possible remedy
FX0-GCAN does not supply any data. PWR ● Red NS ● Red MS ● Red	Supply voltage dip.	Check supply voltage. Reset Flexi Soft system
FX0-GCAN does not supply any data. PWR ● Green NS ☼ Green (1 Hz) MS ☼ Green (1 Hz)	False Node ID/CANopen address. False baud rate (CAN transceiver could be in passive error), FX0-GCAN is in Idle mode.	Check and correct the address. Check and correct the baud rate.
FX0-GCAN does not supply any data. PWR ● Green NS ● Red MS ☼ Red/green	False baud rate and FX0-GCAN transceiver is in bus off state (hardware problem at CAN physical layer). CAN cable has been disconnected.	Check and correct the baud rate. Check CANopen cabling. Reset Flexi Soft system.
FX0-GCAN does not supply any data. PWR ● Green NS ☼ Green (1 Hz) MS ● Green	CANopen master is in stop or pre-operational mode. During initialization of the bus system, another slave could not be initialized. CANopen state of FX0-GCAN is pre-operational. False Node ID/CANopen address.	Set CANopen master into Run state (CANopen state Operational). Check whether all slaves on the bus are “On”. Check CANopen cabling. Check whether CAN master starts automatically. Check and correct the CANopen address.
FX0-GCAN does not supply any data. PWR ● Green NS ● Red MS ● Green	FX0-GCAN transceiver is in error passive. CAN cable has been disconnected.	Check CANopen cabling. Check the diagnostics messages with Flexi Soft Designer. Reset Flexi Soft system.
FX0-GCAN does not supply any data. PWR ● Green NS ☼ Red (1 Hz) MS ☼ Red/green	Node guarding or heartbeat consumer failure. Guarding configuration has been changed.	Check CANopen cabling. Check life guarding time (life time factor ≥ 1) Check heartbeat consumer time (should be ≥ 1.5 × heartbeat producer time) Check the diagnostics messages with Flexi Soft Designer. Reset Flexi Soft system.
FX0-GCAN is in Critical fault mode. PWR ● Green NS ● Red MS ☼ Red (2 Hz)	FX0-GCAN internal device error. CPU firmware version does not support Flexi Soft gateways.	Switch off the power supply of the Flexi Soft system and switch it on again. Check the diagnostics messages with Flexi Soft Designer. Use a FX3-CPUx with the required firmware version (see section 2.2 “Correct use” on page 10). If the error remains, replace the gateway.
FX0-GCAN/Flexi Soft system is in Critical fault mode PWR ● Red NS ○ Off MS ● Red	FX0-GCAN is not plugged properly into the other Flexi Soft module. Module connecting plug is soiled or damaged. Other Flexi Soft module has internal critical error.	Plug the FX0-GCAN in correctly. Clean the connecting socket/plug. Repower the system. Check the other Flexi Soft modules.

6.3 DeviceNet gateway

The following Flexi Soft gateway can be used for DeviceNet: FX0-GDEV.

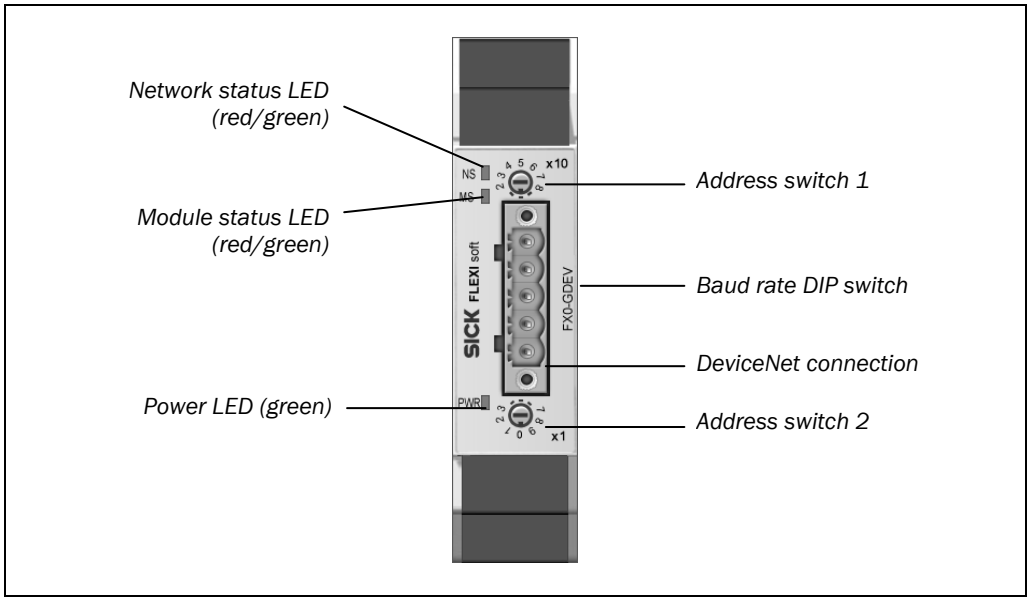
6.3.1 Characteristic of the DeviceNet implementation

- group 2 server (fragmented)
- acknowledge handling
- I/O Messaging (Polled or Change of State/Cyclic) also fragmented
- baud rate adjustable via DIP switch (125 kbit/s, 250 kbit/s, 500 kbit/s)

6.3.2 Interfaces and operation

Controls and status indicators

Fig. 74: Controls and status indicators FX0-GDEV



Tab. 133: Meaning of the status LEDs of the FX0-GDEV

Symbol description:
○: LED is off
● Green: LED lights up green
☼● Red: LED flashes red

LED		Meaning
PWR (power)	○	No voltage supply
	● Green	Operational, voltage supply switched on
	● Red	System error
NS (Network Status)	○	Duplicate MAC ID check in progress
	● Green	Operational
	☼● Green	No connection to the master
	● Red	Duplicate MAC ID check failed
	☼● Red (1 Hz)	Connection timeout
MS (Module Status)	○	Power-up
	● Green	Executing, FLEXBUS+ and process data status: all “Good”
	☼● Green	Idle (cable not attached)
	☼● Red/green	Executing; FLEXBUS+ and process data status: at least one status “Bad”
	● Red	Critical fault, caused by emergency bit
	☼● Red (1 Hz)	Configuration required or in progress
	☼● Red (2 Hz)	Critical fault, caused by gateway itself

For diagnostics see section 6.3.7 “Diagnostics and troubleshooting” on page 171.

How to set the DeviceNet address via hardware address switches:

➢ Set the DeviceNet address using the hardware address switches on the device front. Then switch the Flexi Soft system off and back on again.

Tab. 134: Address switches on the FX0-GDEV

Switch	Function
× 10	Address switch 1 10 position rotary switch for setting the module address (ones)
× 1	Address switch 2 10 position rotary switch for setting the module address (ones)

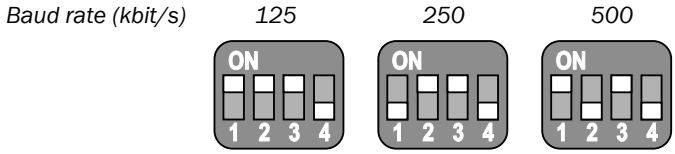
Notes

- The module addresses that can be selected using the address switches are in the range 1-63.
- If both address switches on the device are set to 0, then the address configured in Flexi Soft Designer is used. There the address can be set in the range 0-63.
- If a higher address is selected, address 63 is active.

How to set the baud rate via hardware DIP switches:

➢ Set the baud rate using the DIP switches on the device. Then switch the Flexi Soft system off and back on again.

Fig. 75: DIP switch settings on the FX0-GDEV



Tab. 135: DIP switch settings on the FX0-GDEV

Baud rate [kbit/s]	DIP 1	DIP 2	DIP 3	DIP 4
125	On	On	On	Off
250	Off	On	On	Off
500	On	Off	On	Off

Notes

- If all DIP switches are set to Off, then the setting for the baud rate in Flexi Soft Designer is used.
- All other DIP switch settings will set the baud rate to 125 kbit/s.

How to set the DeviceNet address and baud rate via software using the Flexi Soft Designer:

➢ Set both address switches to 0.

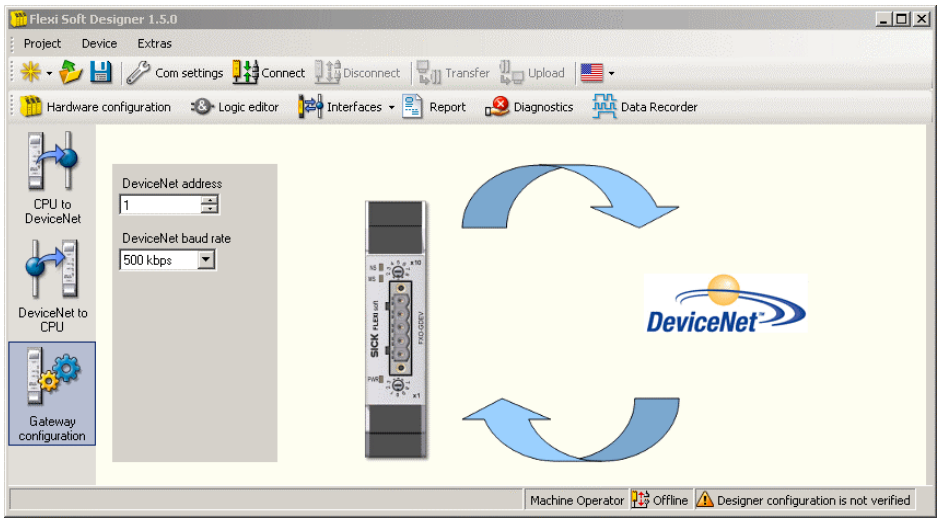
➢ Set all DIP switches to Off.

➢ Open the Flexi Soft Designer and load the hardware configuration including the DeviceNet gateway. Ensure that your project is offline.

➢ Click on the **Gateways** button above the main window and select the FX0-GDEV or double click the FX0-GDEV in the hardware configuration view to open the gateway configuration dialog.

➢ Click on **Gateway configuration** on the left hand menu. The following dialog will be displayed:

Fig. 76: Setting the DeviceNet address of the FX0-GDEV



- Select the DeviceNet address in the **DeviceNet address** field.
- Select the baud rate in the **DeviceNet baud rate** field.
- Click **Connect** to go online and transfer the configuration to the Flexi Soft system.

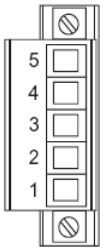
Notes

- The address that can be set via the hardware address switch ranges from 1-63.
- The address that can be set via the Flexi Soft Designer software ranges from 0-63.
- The DeviceNet master cannot overwrite the address.
- If the DeviceNet address and the baud rate are set using the Flexi Soft Designer, the settings will become effective immediately after transferring the configuration (i.e. withut switching off and switching on the Flexi Soft system). Exception: If the system is in the Bus Off state, then you must switch off and on again the device for the changes to become effective.

Plug assignment

The connection to the DeviceNet fieldbus is made using a 5-pin open style connector.

Fig. 77: Open style connector and assignment FX0-GDEV



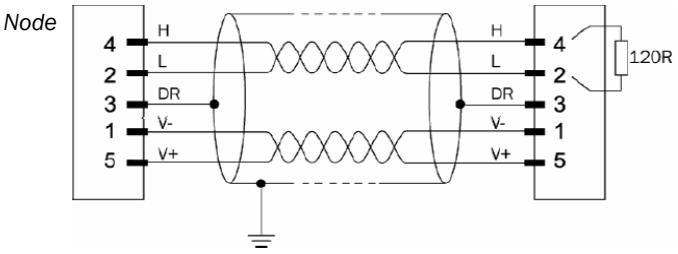
Pin	Description	
5	V+ (24 V)	Voltage supply
4	H CAN_H	DeviceNet High
3	DR (CAN_SHLD)	Screen connection (optional)
2	L CAN_L	DeviceNet Low
1	V– (GND)	GND/0V

Bus cable

DeviceNet is based on CAN and is therefore a 2-wire bus system to which all bus users are connected in parallel. Short stubs are also possible. The signal wires H and L must be terminated at both ends of the bus using a 120 Ω resistor. The screen must be continuous over the entire length of the bus and earthed at one point. It is recommended to use a twisted-pair cable with two twisted, screened pairs of cores. The 24 V supply voltage is connected to the second pair of cores.

The transmission rate, depending on the network length, is between 125 kbit/s and 500 kbit/s. The possible network lengths are 100 m at 500 kbit/s up to 500 m at 125 kbit/s.

Fig. 78: DeviceNet bus cable



Depending on the cable used and the baud rate set, the following maximum physical values are possible:

Tab. 136: Maximum cable lengths FX0-GDEV

Cable length	125 kbit/s	250 kbit/s	500 kbit/s
Overall length with thick cable (≥ 0,34 mm²)	500 m	250 m	100 m
Overall length with thin cable	100 m	100 m	100 m
Overall length with ribbon cable	380 m	200 m	75 m
Maximum stub length	6 m	6 m	6 m
Maximum length of all stubs	156 m	78 m	39 m

6.3.3 Setting up DeviceNet communication

To setup the DeviceNet communication between the gateway and the higher level functional logic, the following steps must be taken:

- Install EDS file.
- Define type of communication.
- Define contents of the communication.
- Select and use data mapping.

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

The device characteristics are described using the Electronic Data Sheet (EDS), which every standard bus configuration tool uses. Before the FX0-GDEV can be used as device in the network configuration tool for the first time, the EDS file of the gateway must be installed into the hardware catalogue of the tool.

You will find the EDS file and device icon for PLC interfacing ...

- in the Internet on the FX0-GDEV product page.
- in the Flexi Soft Designer program folder on your hard disk (default installation folder is “C:\programs\SICK\FlexiSoft\DeviceDescriptions\...”).

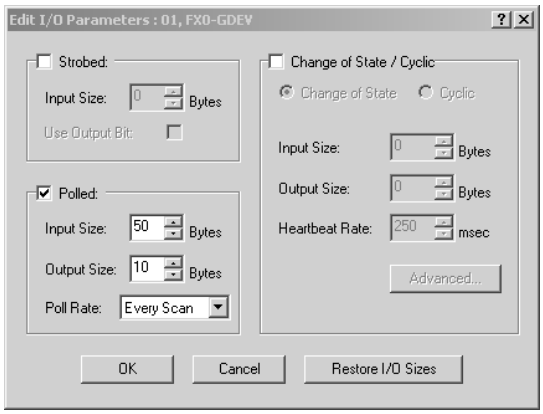
➢ Download the EDS file and device icon from www.sick.com, on the FX0-GDEV product page.

➢ Follow the instructions in the online help or in the user manual of the DeviceNet configuration tool for installing EDS files.

Step 2: Define the type of communication

➢ Choose the type of communication that is to be used between the gateway and the higher level functional logic (Polled or Change of state/Cyclic).

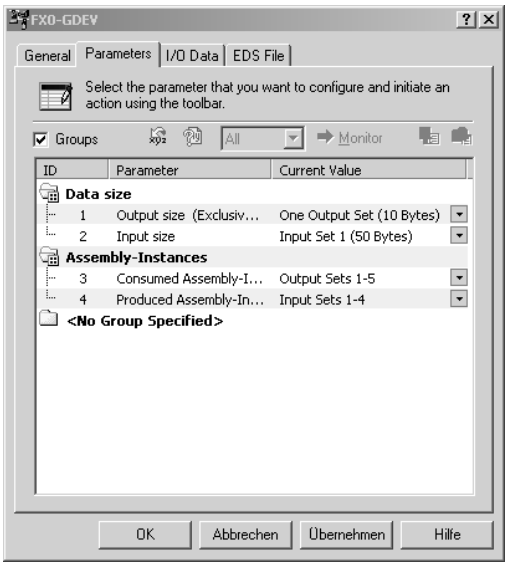
Fig. 79: Configuration of the type of communication based on the example of the DeviceNetManager™ from Allen Bradley



Step 3: Define the contents of the communication

➢ Choose the input and output data sets that are to be transmitted between the gateway and the functional logic.

Fig. 80: Selection of input data set 1 and output data set 1 using DeviceNetManager™ from Allen Bradley



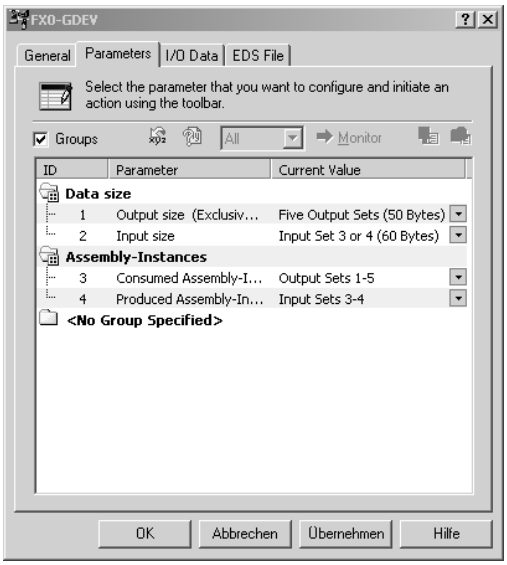


Fig. 81: Selection of input data set 3 and output data set 1-5 using DeviceNetManager™ from Allen Bradley

Step 4: Configure data mapping

For information how to define and customize the content and tag names of the input and output data sets please see chapter 7 “Layout and content of the process image” on page 173 and the operating instructions for the Flexi Soft Designer software (SICK part no. 8012998).

6.3.4 Supported DeviceNet features

The FX0-GDEV supports the following functions of a DeviceNet Group 2 slave (server):

- Explicit Messaging (fragmented)
- Implicit Messaging (I/O messages)
  - Poll I/O (fragmented)
  - Change of state I/O (fragmented)
  - Cyclic I/O (fragmented)
- Unconnected Explicit Request Messages (pre-defined master/slave connection set)
- Device heartbeat and shutdown messages
- UCMM port (Unconnected Message Manager)
- Duplicate MAC ID messages
- Offline Connection Set

Pre-defined master/slave connection set

The FX0-GDEV supports a pre-defined master/slave connection set that makes it possible to establish a DeviceNet connection that requires fewer network and device resources. This set contains an explicit messaging connection and makes possible several different I/O connections, including Poll, COS and Cyclic.

Poll I/O messages

The Poll command is sent by the master. A Poll command is directed at a single, specific slave (point-to-point connection). A master must send a dedicated Poll command to each slave that is to be polled. On the receipt of the Poll command the slave replies to the master with an I/O message.

Change of state I/O and Cyclic I/O messages

A Change of state message is sent by either the master or the slave. Change of state and Cyclic messages are directed at an individual, specific slave (point-to-point connection). They can be answered with a confirmation.

Explicit messages

Explicit message requests are used to read or write attributes. The result of such a request is signaled with an explicit message reply.

Unconnected Messages

Unconnected messages (UCMM messages) are used to establish or close explicit connections between two devices. They are processed by the Unconnected Message Manager (also termed a UCMM port). It is possible to establish a maximum of three simultaneous connections via UCMM.

Group 2 Only Unconnected Explicit Request Messages

Group 2 Only Unconnected Explicit Request messages are used to allocate or close the pre-defined master/slave connection set. They are only available on devices that do not support a UCMM port and are used as an alternative method to establish a connection. However only one connection with a single partner is possible, specifically the master.

Duplicate MAC ID messages

These messages are used to open the network access state machine that prevents two or more nodes having an identical MAC ID in the same network.

6.3.5 DeviceNet protocol settings

Assembly object

Tab. 137: Assembly object

Parameter name	Meaning	Value
SUPPORT_ASSEMBLY_ATTRIB_1N2	0 = Attributes 1 and 2 of the assembly object are not supported 0 = Attributes 1 and 2 of the assembly object are supported	1
SUPPORT_ASSEMBLY_ATTRIB_4	0 = Attribute 4 of the assembly object is not supported 0 = Attribute 4 of the assembly object is supported	1
ASMOBJ_NUM_OF_INSTANCES	Number of assembly instances	9

6.3.6 Assemblies

Assembly objects are used to exchange input or output data that comprise more than one attribute via a single connection. Here a data package is prepared that can be referenced as attribute 3 of the Assembly object class (class 4).

The information on which attributes are combined into an assembly object is called assembly mapping.

Produced Assembly Instance (Target—Originator)

Tab. 138: Produced Assembly Instance

Instance ID	Description	Size [bytes]	Received data
1	Input data set 1-4	50 82 142 202	Input data set 1 Input data set 1-2 Input data set 1-3 Input data set 1-4
2	Input data set 2-4	32 92 152	Input data set 2 Input data set 2-3 Input data set 2-4
3	Input data set 3-4	60 120	Input data set 3 Input data set 3-4
4	Input data set 4	60	Input data set 4

Notes

- All values are data type **Array of USINT**. Therefore the possible values range from 0-255.
- The access rule for all instances is **GET**.
- All these assemblies can be accessed via Implicit Message or via Explicit Message.

Consumed Assembly Instance (Originator—Target)

Tab. 139: Consumed Assembly Instance

Instance ID	Description	Size [bytes]	Sent Data
5	Output data sets 1-5	10 20 30 40 50	One output data set Two output data sets Three output data sets Four output data sets Five output data sets
6	Output data sets 2-5	10 20 30 40	One output data set Two output data sets Three output data sets Four output data sets
7	Output data sets 3-5	10 20 30	One output data set Two output data sets Three output data sets
8	Output data sets 4-5	10 20	One output data set Two output data sets
9	Output data set 5	10	One output data set

Notes

- All values are data type **Array of USINT**. Therefore the possible values range from 0-255.
- The access rule for all instances is **GET/SET**.
- All these assemblies can be accessed via Implicit Message or via Explicit Message.

Individual Data Set Transfer object

(73h — one instance per data set)

The vendor specific **Individual Input Data Set Transfer** object defines the attributes by which the PLC can request either full input data sets or individual parameters within an input data set.

Class attributes

Tab. 140: Class attributes for the Individual Data Set Transfer object (73h)

Attribute ID	Name	Data type	Data value(s)	Access rule
1	Revision	UINT	1	Reading
2	Max. instance	UINT	4	Reading
3	Number of instances	UINT	4	Reading

Instance attributes

Tab. 141: Instance attributes for the Individual Data Set Transfer object (73h)

Attribute ID	Name	Data type	Data value(s)	Access rule
1 to *n* (dependent on data set definition)	Request input data set specific data	Dependent on data set definition	0-255	Reading

General services

Tab. 142: Individual Input Data Set Transfer object (73h) general services

Service code	Implemented in Class	Implemented in Instance	Service name
01h	Yes	Yes	Get_Attributes_All
0Eh	Yes	Yes	Get_Attribute_Single

Instance attribute definitions

Attribute 1 to *n* — Request input data set specific parameters
These attributes return the input data set specific data arrays. **Get Attribute Single** requests for a specific input data set only return the parameter information for the requested data set. **Get Attribute All** requests return the entire data set.

The data set attributes, numbered from 1 to N, refer to each individual attribute of each individual input data set. Each instance relates to a unique input data set and each input data set has a unique attribute numbering scheme. The following tables reflect the attribute definitions for each input data set.

Get All Data Set Attributes request

All data set information will be returned in integer (16 bit word) format. For byte data, the first byte will be placed in the least significant or rightmost byte of the integer and the second byte will be placed in the most significant or leftmost byte of the integer.

Example:

For an input data set, the data will be returned as follows:

- IntegerArray[0]: BBAAh — AA = BYTE1; BB = BYTE2
- IntegerArray[1]: DDCCh — CC = BYTE3; DD = BYTE4
- ...
- IntegerArray[6]: NNMMh — MM = BYTE13; NN = BYTE14

Instance 1 — Input data set 1 attribute definitions

Tab. 143: Individual Input Data Set Transfer object (73h) instance 1 attribute definitions

Attribute number	Data set parameter	Size
1	Byte 1	SINT
2	Byte 2	SINT
...	...	...
50	Byte 50	SINT

Instance 2 — Input data set 2 attribute definitions

Tab. 144: Individual Input Data Set Transfer object (73h) instance 2 attribute definitions

Attribute number	Data set parameter	Size
1	Overall CRC	UDINT
2	System CRC (SCID)	UDINT
3	Reserved	UDINT
4	Reserved	UDINT
5	Reserved	UDINT
6	Reserved	UDINT
7	Reserved	UDINT
8	Reserved	UDINT

Instance 3 — Input data set 3 attribute definitions

Tab. 145: Individual Input Data Set Transfer object (73h) instance 3 attribute definitions

Attribute number	Data set parameter	Size
1	Module status module 0	UINT[2]
2	Module status module 1	UINT[2]
...	...	...
15	Module status module 14	UINT[2]

Instance 4 — Input data set 4 attribute definitions

Tab. 146: Individual Input Data Set Transfer object (73h) instance 4 attribute definitions

Attribute number	Data set parameter	Size
1	Reserved	UINT[2]
2	Reserved	UINT[2]
...	...	...
15	Reserved	UINT[2]

6.3.7 Diagnostics and troubleshooting

For information on how to perform diagnostics on the Flexi Soft system please refer to the operating instructions for the Flexi Soft Designer software (SICK part no. 8012998).

Tab. 147: Troubleshooting for the FX0-GDEV

Symbol description:
○: LED is off
● Green: LED lights up green
☼ Red: LED flashes red

Error	Possible cause	Possible remedy
FX0-GDEV does not supply any data. PWR ● Green NS ○ Off MS ☼ Red (1 Hz)	Configuration required, node guard or heartbeat message have not been sent. Configuration download is not completed.	Configure the FX0-GDEV and download the configuration to the device. Wait until the configuration download has been completed.
FX0-GDEV does not supply any data. PWR ● Green NS ● Green MS ☼ Red (1 Hz)	Configuration download is not completed.	Wait until the configuration download has been completed.
FX0-GDEV does not supply any data. PWR ● Green NS ● Green MS ☼ Red/green	No data transmission since power-up	Start the data transmission.
FX0-GDEV does not supply any data. PWR ● Green NS ☼ Green MS ☼ Red/green	No data transmission since power-up False baud rate False Node ID/DeviceNet address The cable has been disconnected.	Start the data transmission. Check and correct the baud rate. Check and correct the Node ID and DeviceNet address. Check cabling.
FX0-GDEV does not supply PDO data. PWR ● Green NS ○ Off/● Red/● Green MS ☼ Green (1 Hz)	FX0-GDEV is in Idle mode. Node guard or heartbeat messages will be sent. Flexi Soft configuration is not verified and main module is stopped.	CPU/application is stopped. Start CPU (change into Run state). Verify the configuration with Flexi Soft Designer and start the main module.
FX0-GDEV does not supply PDO data. PWR ● Green NS ● Green MS ○ Off	Supply voltage too low.	Check supply voltage.
FX0-GDEV does not supply any data. PWR ● Red NS ● Red MS ● Red	Supply voltage dip.	Check supply voltage. Reset Flexi Soft system.
FX0-GDEV does not supply any data. PWR ● Green NS ☼ Green (1 Hz) MS ☼ Green (1 Hz)	False Node ID/DeviceNet address. False baud rate, FX0-GDEV is in Idle mode.	Check and correct the Node ID and DeviceNet address. Check and correct the baud rate.
FX0-GDEV does not supply any data. PWR ● Green NS ● Red MS ☼ Red/green	False baud rate and FX0-GDEV transceiver is in bus off state (hardware problem at DeviceNet physical layer). The cable has been disconnected.	Check and correct the baud rate. Check cabling. Reset Flexi Soft system.
FX0-GDEV does not supply any data. PWR ● Green NS ☼ Green (1 Hz) MS ● Green	DeviceNet master is in stop or pre-operational state. During initialization of the bus system, another slave could not be initialized. DeviceNet state of the FX0-GDEV is pre-operational. False Node ID/DeviceNet address	Set DeviceNet master into Run state (DeviceNet state Operational). Check whether all slaves on the bus are “On”. Check cabling. Check whether DeviceNet master starts automatically. Check and correct the DeviceNet address.
FX0-GDEV does not supply any data. PWR ● Green NS ● Red MS ● Green	FX0-GDEV transceiver is in error passive. The cable has been disconnected.	Check cabling. Check the diagnostics messages with the Flexi Soft Designer. Reset Flexi Soft system.

Error	Possible cause	Possible remedy
FX0-GDEV does not supply any data. PWR ● Green NS ☼ Red (1 Hz) MS ☼ Red/green	Node guarding or heartbeat consumer failure Guarding configuration has been changed.	Check cabling. Check life guarding time (life time factor ≥ 1). Check heartbeat consumer time (should be ≥ 1.5 × heartbeat producer time). Check the diagnostics messages with the Flexi Soft Designer. Reset Flexi Soft system.
FX0-GDEV is in Critical fault mode. PWR ● Green NS ● Red MS ☼ Red (2 Hz)	FX0-GDEV internal device error CPU firmware version does not support Flexi Soft gateways.	Switch off the power supply of the Flexi Soft system and switch it on again. Check the diagnostics messages with the Flexi Soft Designer. Use a FX3-CPUx with the required firmware version (see section 2.2 “Correct use” on page 10). If the error remains, replace the gateway.
FX0-GDEV/Flexi Soft system is in Critical fault mode. PWR ● Red NS ○ Off MS ● Red	FX0-GDEV is not plugged properly into the other Flexi Soft modules. Module connecting plug is soiled or damaged. Other Flexi Soft module has internal critical error.	Plug the FX0-GDEV in correctly. Clean the connecting socket/plug. Repower the system. Check the other Flexi Soft modules.

7 Layout and content of the process image

7.1 Routing

The process image transmitted by the Flexi Soft gateways into the network consists of the operational data (e.g. logic results, input and output states) and the diagnostics data (e.g. module status, CRCs). These data are organised in 4 data sets.

Tab. 148: Content of the data sets 1-4

Data set	Content	Size	Can be customized
1	Operational data	Max 50 bytes[16)]	Yes
2	CRCs	32 bytes	No
3	Status and diagnostics	60 bytes	No
4	Reserved	60 bytes	No

The operational data in data set 1 are organized into one or several data blocks, dependent of the network protocol. For detailed information about the modularisation of the data sent into the network please see Tab. 149 and read the chapter on the related gateway.

The content of data set 1 can be freely customized with a granularity of 1 byte but is preconfigured in the delivery status (see section 7.2 “Default settings for the operational data” on page 174 and section 7.3 “Customizing the operational data (Flexi Soft to Network)” on page 175).

The diagnostics data in data sets 2-4 depend on the network protocol used and are described in the chapter on the related gateway.

16) FX0-GCAN: 32 Bytes.

7.2 Default settings for the operational data

In the delivery status, the operational data are pre-configured. Depending on the gateway used, these data are subdivided in several data blocks.

The following table gives an overview which bytes are assigned to the default configuration and how the data are modularised for the different gateways.

Tab. 149: Default configuration for the operational data transmitted into the network

	EtherNet/IP, Modbus TCP, Ethernet TCP/IP		**PROFINET IO, PROFIBUS DP**	
Byte	**Default assignment**	**Input data set**	**Default assignment**	**Input data block**
0	Logic result 0	#1 (50 bytes)	Module 1 input	#1 (12 bytes)
1	Logic result 1		Module 2 input	
2	Logic result 2		Module 3 input	
3	Logic result 3		Module 4 input	
4	Module 1 input		Module 5 input	
5	Module 2 input		Module 6 input	
6	Module 3 input		Module 7 input	
7	Module 4 input		Module 8 input	
8	Module 5 input		Module 9 input	
9	Module 6 input		Module 10 input	
10	Module 7 input		Module 11 input	
11	Module 8 input		Module 12 input	
12	Module 9 input		Module 1 output	#2 (12 bytes)
13	Module 10 input		Module 2 output	
14	Module 11 input		Module 3 output	
15	Module 12 input		Module 4 output	
16	Module 1 output		Module 5 output	
17	Module 2 output		Module 6 output	
18	Module 3 output		Module 7 output	
19	Module 4 output		Module 8 output	
20	Module 5 output		Module 9 output	
21	Module 6 output		Module 10 output	
22	Module 7 output		Module 11 output	
23	Module 8 output		Module 12 output	
24	Module 9 output		Logic result 0	#3 (12 bytes)
25	Module 10 output		Logic result 1	
26	Module 11 output		Logic result 2	
27	Module 12 output		Logic result 3	
28	Gateway direct output values 0		Gateway direct output values 0	
29	Gateway direct output values 1		Gateway direct output values 1	
30	Gateway direct output values 2		Gateway direct output values 2	
31	Gateway direct output values 3		Gateway direct output values 3	
32-35	Not assigned		Not assigned	
36-47	Not assigned		Not assigned	#4 (12 bytes)
48-49	Not assigned		Not assigned	#5 (2 bytes)

For the FX0-GETC please see Tab. 75 on page 107.

For the FX0-GCAN please see Tab. 110 on page 143.

The default byte assignment can be freely customized as described in the following section.

7.3 Customizing the operational data (Flexi Soft to Network)

This section outlines briefly how you can customize the operational data that the Flexi Soft gateway transmits to the network. You will find more detailed information in the Flexi Soft Designer software operating instructions (SICK part no. 8012998).

In the delivery status, the data routing configuration of the Flexi Soft gateways is shown in the gateway configuration dialog.

- Click on the **Gateways** button above the main window and select the respective gateway or double click the desired gateway in the hardware configuration view to open the gateway configuration dialog.
- Click on the **CPU to network** tab on the left hand menu to display the routing configuration dialog.

The default setting is as follows (example for Modbus TCP):

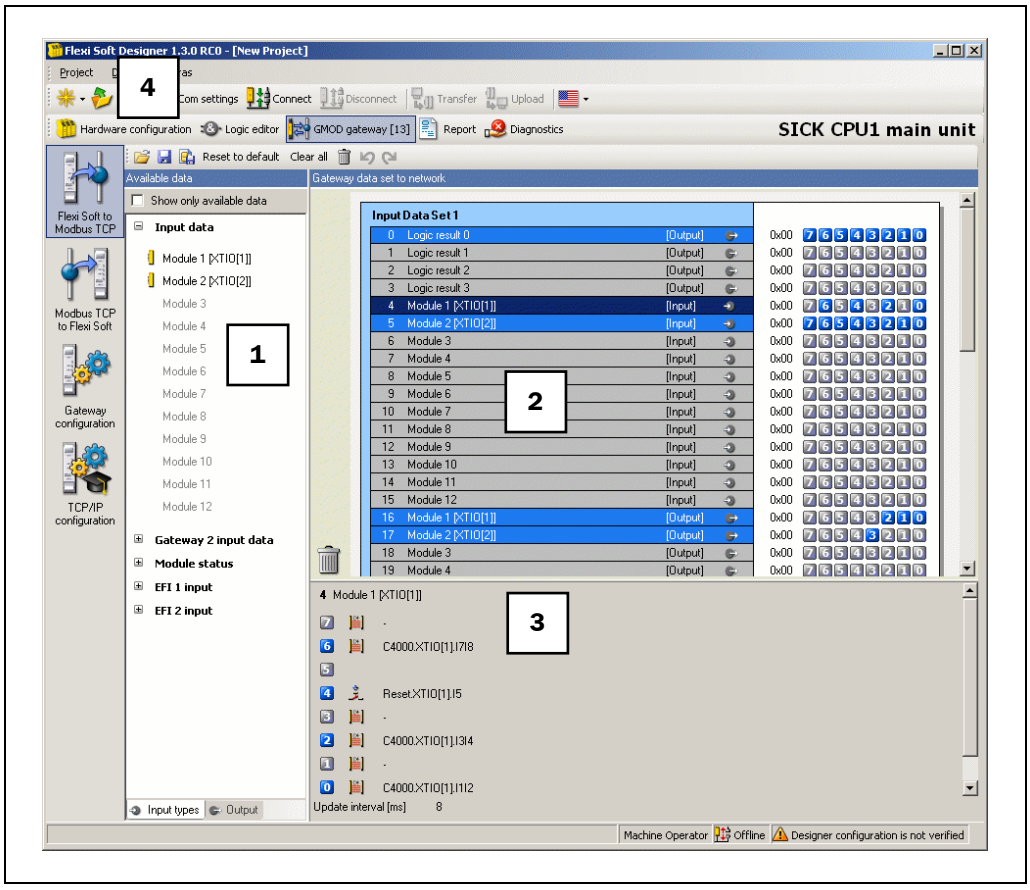


Fig. 82: Default configuration for the operational data transmitted into the network

Basically this dialog is divided into three areas: **Available data** [1], **Gateway data set to network** [2] and **Tag names** [3]. The upper left corner of the dialog holds the toolbar [4].

7.3.1 The toolbar

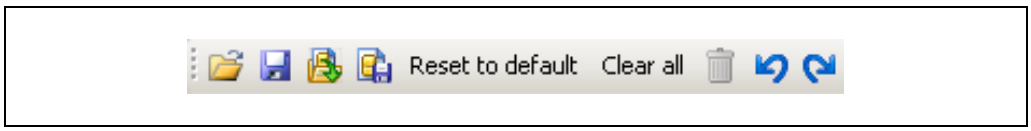


Fig. 83: Toolbar for the routing configuration

The toolbar contains buttons for the following actions (from left to right):

- The **Load user configuration** and **Save user configuration** buttons allow you to load and/or save a configuration including the used tag names in XML format. If you load a configuration, all previously made changes that have not been saved will be lost. You can not undo this action.
- With the **Import** and **Export** buttons you can import and export the used tag names as a CSV file or in a network specific file format, e.g. SIEMENS .seq for PROFIBUS or PROFINET. This allows you to import and use the assigned tag names in a PLC program.

Note The **Import** button is only available for the *Network to gateway* routing configuration.

- **Reset to default** restores the default routing configuration. You will be prompted for confirmation. If you click **Yes**, all previously made changes that have not been saved will be lost. You can not undo this action.
- **Clear all** clears the configuration, i.e. deletes all assigned bytes in the **Gateway data set to network** area. You will be prompted for confirmation.
- **Delete routing** deletes the selected byte from the **Gateway data set to network** area.
- The **Undo** and **Redo** buttons allow you to undo or redo changes you made to your configuration.

7.3.2 Available data area

This area offers all sources from which data may be routed into the network. It is divided in two views holding the available **Input types** and **Output** data. You can switch between these views using the file cards at the bottom.

- The **Input types** view contains the input values for the connected Flexi Soft modules and EFI devices. If your Flexi Soft system contains a second gateway, the input data of this gateway (i.e. data received from the network the second gateway is connected to) will be available here as well.
- The **Output** view offers the output values for the connected Flexi Soft modules and EFI devices as well as the **Logic results** from the logic editor.

All sources supported by the current configuration are displayed in black:

- connected Flexi Soft modules
- connected EFI devices
- configured logic results[17)]
- gateway input data available from another gateway in the system

Sources currently not configured will be displayed in grey. Activating the **Show only available data** checkbox in the upper left corner hides the unused sources from the view.

Sources that offer “live” data are marked with a little icon next to the text.

[17)] In the default configuration, only the first logic result byte (Logic result 0) is active and available. You can activate more logic result output bits in the logic editor (see the Flexi Soft software operating instructions, SICK part no. 8012998).

How to add a data byte to the routing table:

- Drag and drop an element (i.e. byte) from the **Available data** area to a free slot in the **Gateway data set to network** area. If the desired position is not free, you will have to clear it first by deleting or moving the byte currently assigned to it.

Note It is possible to use the same byte several times in the routing table.

7.3.3 Gateway data set to network area

This area contains the routing table. It shows the current content of the Flexi Soft gateway's input data modules, bytes and bits highlighted blue will hold "live" data from the system since the hardware configuration does support the source. Bytes highlighted grey actually do not have data associated with them since the hardware configuration does not support the sources.

How to delete a data byte from the routing table:

- Drag and drop the byte you want to delete to the trashcan icon in the bottom left corner of the **Gateway data set to network** area.

Or:

- Select the byte you want to delete by clicking it with the left mouse button. Then, click on the **Delete routing** button in the toolbar.

Or:

- Call up the context menu by clicking the respective byte with the right mouse button. In the context menu, select the **Delete routing** command.

How to move a data byte to another place in the routing table:

- Drag and drop the byte you want to move to the desired position. If the desired position is not free, you will have to clear it first by deleting or moving the byte currently assigned to it.

7.3.4 Tag names area

This area shows the tag names associated with each bit of the byte currently selected in the **Available data** or the **Gateway data set to network** area. You can enter these tag names in the logic editor and in the hardware configuration dialog (e.g. for extension modules).

In the **Tag names** area of the Flexi Soft to Network configuration dialog, it is not possible to edit tag names, with the exception of the gateway direct outputs (see the following section).

7.3.5 Gateway direct output values

It is possible to write values directly from the logic editor into a gateway. In the default process image, four bytes are reserved for these gateway direct output values which can be found in the logic editor in the **Outputs** file card.

Note In order to use gateway direct output values, a CPU with firmware V2.00.0 or higher is required.

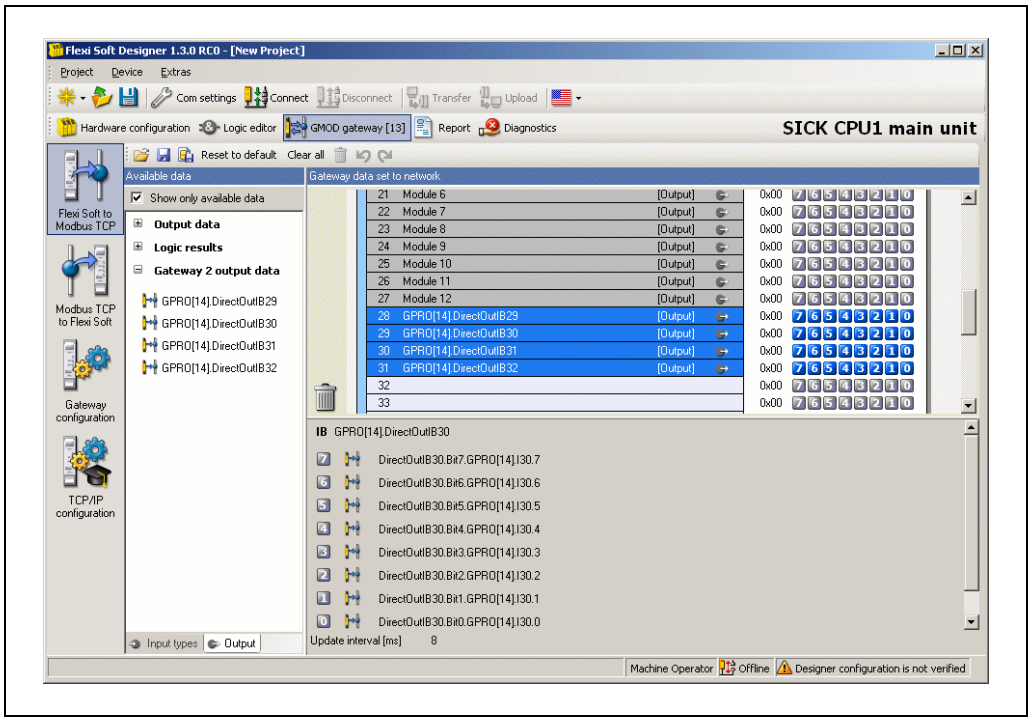


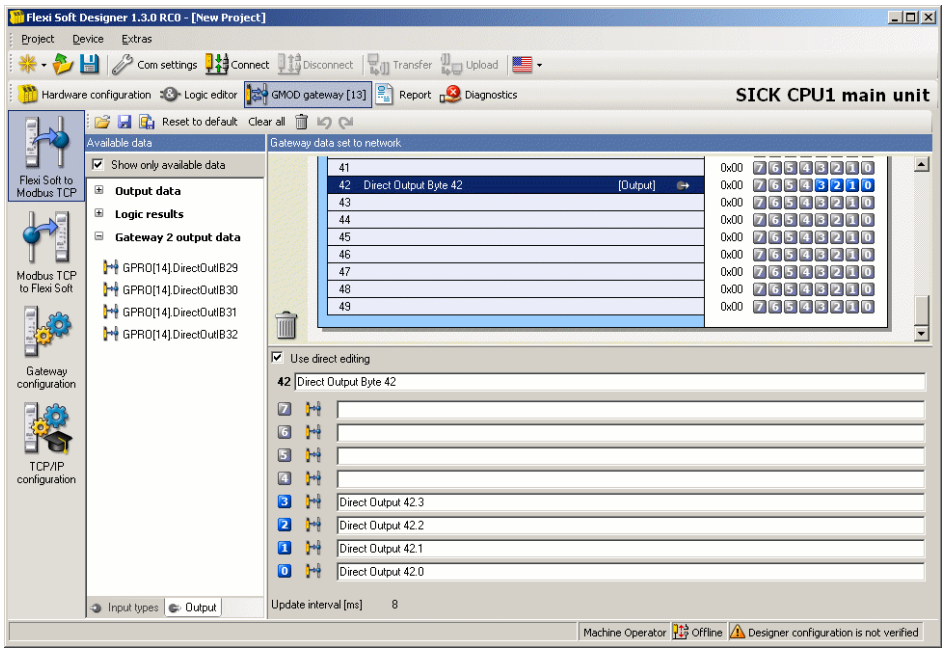
Fig. 84: Gateway direct output values in the default process image

If required, you can configure any byte as gateway direct output value. In order to do this, you need to assign tag names for the bits you want to use.

How to define additional gateway direct output values:

- Click on a free byte in the **Gateway data set to network** area to select it.
- Activate the **Use direct editing** checkbox in the upper left corner of the **Tag names** area. You will now be able to edit the tag names for this byte.
- If desired, enter a tag name for the selected byte.
- Enter tag names for the individual bits of the selected byte that you want to use as gateway direct output values.

Fig. 85: Defining additional gateway direct output values in the process image



All bits of the selected byte with a tag name will now appear in the logic editor in the **Outputs** file card.

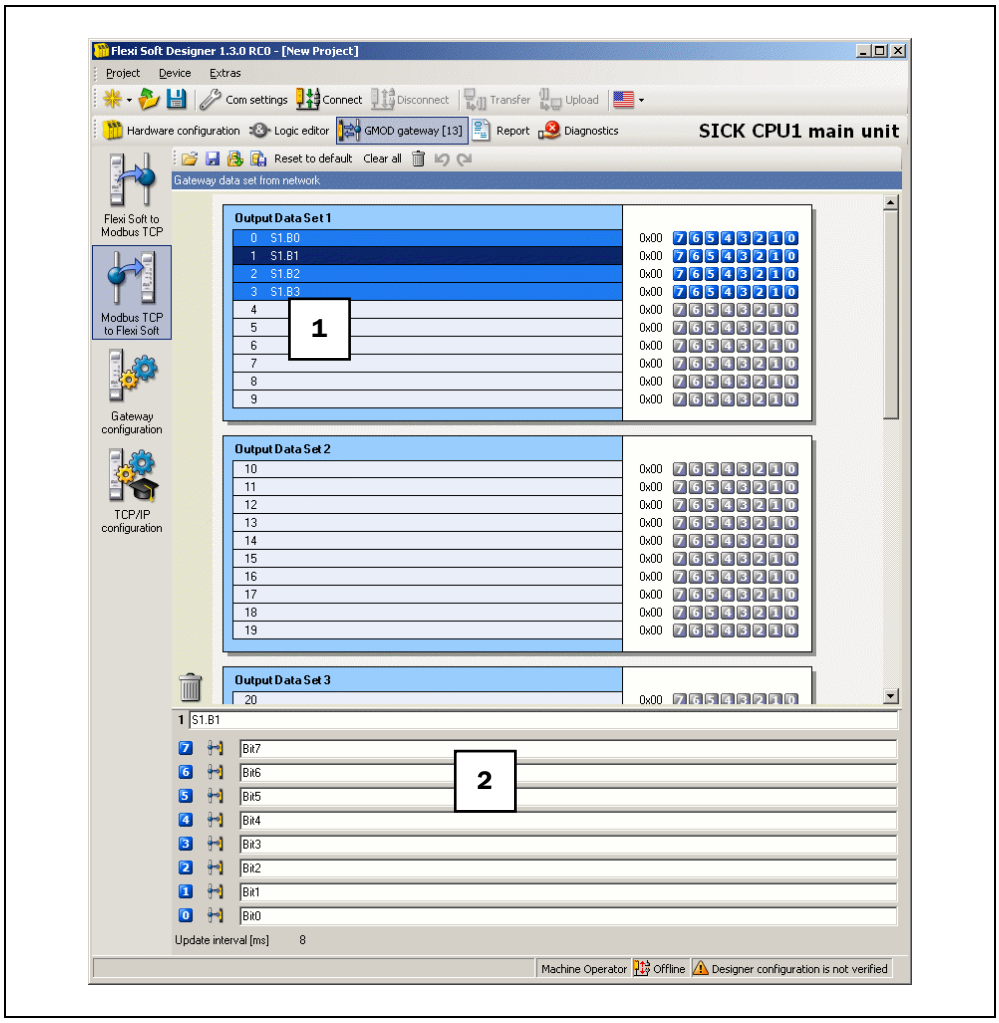
Note You can edit the predefined gateway direct output values in the same way.

7.3.6 Output data configuration (Network to Flexi Soft)

In order to enable incoming data bits:

➢ Click on **Network to CPU** on the left hand menu. The following dialog appears:

Fig. 86: Network to Flexi Soft dialog of the FX0-GMOD



Basically this dialog is divided into two areas: **Gateway data set from network** [1] and **Tag names** [2]:

The **Gateway data set from network** area shows the current configuration of the output modules.

The **Tag names** area shows the tag names associated to the byte selected in the **Gateway data set from network** area.

➢ Select a byte in the **Gateway data set from network** area.

➢ For each bit of the selected byte that you wish to use, enter a tag name in the **Tag names** area.

Each bit you enter a tag name for here will be available within the logic editor or for the process image of a second gateway:

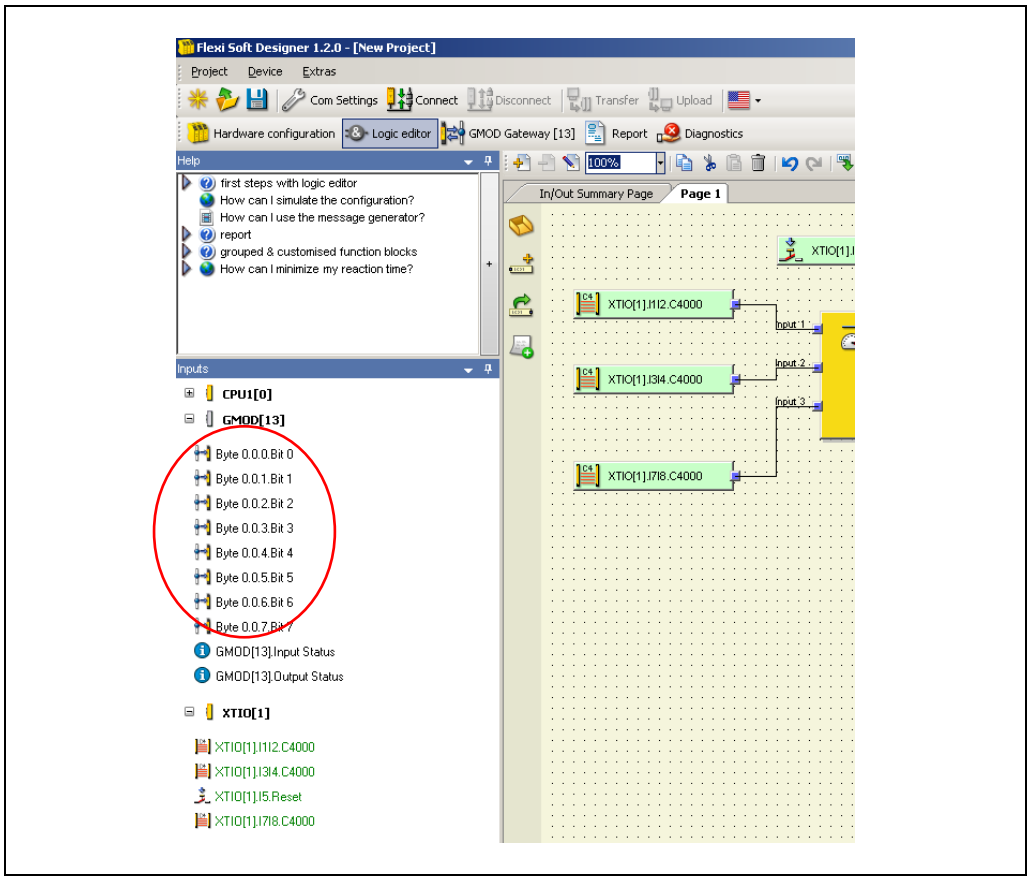


Fig. 87: Tag names of incoming bits in the logic editor dialog of the FX3-CPUx

7.3.7 Saving and loading a configuration

Using the buttons **Load user configuration** and **Save user configuration** you can save and load your configuration in XML format. If you load a configuration, all previously made changes that have not been saved will be lost. You can not undo this action.

7.3.8 Importing and exporting a configuration

With the **Import** and **Export** buttons you can import and export a configuration including the tag names used as a CSV file or in a network specific file format, e.g. SIEMENS .seq for PROFIBUS or PROFINET. This allows you to import and use tag names you have assigned in the Flexi Soft project in the PLC program and vice versa.

If you import a configuration, all previously made changes that have not been saved will be lost. You can not undo this action.

Note The **Import** button is only available for the *Network to gateway* routing configuration.

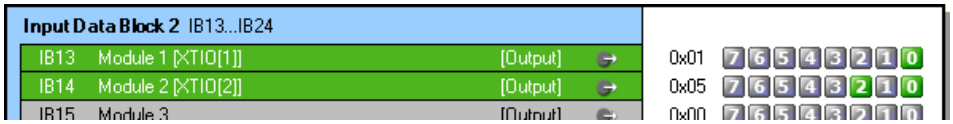
7.4 Monitoring the operational data online

When the Flexi Soft system is online and running, you can monitor the operational data online in the gateway configuration window.

- Click on the **Gateways** button above the main window and select the respective gateway or double click the desired gateway in the hardware configuration view to open the gateway configuration dialog.
- Click on the **Flexi Soft to Network** or the **Network to Flexi Soft** tab on the left hand menu to display the routing view for the input or output data you want to monitor.

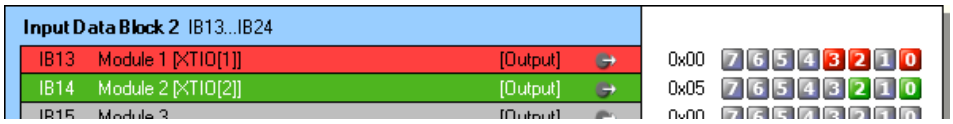
For both directions, **Flexi Soft to Network** as well as **Network to Flexi Soft**, inactive bits are displayed grey while active bits are highlighted green:

Fig. 88: Active and inactive bits in the online process image



In the **Flexi Soft to Network** view, bits that are inactive due to an error are displayed red. This could be the case e.g. for the outputs of an FX3-XTIO module if the power supply of this module is faulty:

Fig. 89: Inactive Network input bits as a result of an error.



In the **Network to Flexi Soft** view, bits that have no tag name assigned (so that they can not be used in the logic editor) but which are included in the process image that the Flexi Soft gateway receives from the PLC, are highlighted yellow:

Fig. 90: Network output bits with no tag names assigned in the online process image

Note The Flexi Soft gateways always reflect the actual physical status of the inputs and outputs on the connected modules and devices. This means that even when Force mode is active and inputs that are physically **Low** are forced **High** (or vice versa) the actual physical status of these inputs will be transmitted to the PLC instead of the (virtual) forced status. If, however, as a result of forcing of one or several inputs, one or several outputs change their status, the changed status of these outputs will be transmitted to the PLC since the actual physical status of the outputs on the devices has changed.

8 Technical specifications

8.1 Technical specifications gateways

8.1.1 EtherNet/IP, Modbus TCP, PROFINET IO

Tab. 150: Technical specifications FX0-GENT, FX0-GMOD and FX0-GPNT

Interface	
Fieldbus	EtherNet/IP, Modbus TCP, PROFINET IO
Integrated switch	3-port layer-2 managed switch with Auto-MDI-X for automatic detection of crossed Ethernet cable
Connection technique	RJ-45 socket
Transfer rate	10 Mbit/s (10 Base-T) or 100 Mbit/s (100 Base-TX), autosensing
Update rate (heartbeat rate)	Configurable from 40 ... 65,535 ms
Change of state (COS) update rate	10 ms
Addressing factory setting	IP: 192.168.250.250 Subnet mask: 255.255.0.0 Default gateway: 0.0.0.0
MAC address	Printed on type label, example: 00:06:77:02:00:A7

8.1.2 EtherCAT

Tab. 151: Technical specifications FX0-GETC

Interface	
Fieldbus	EtherCAT
Connection technique	RJ-45 socket
EtherCAT application cycle time	4 ms
Watchdog Time Process Data	≥ 5 ms
Watchdog Time Process Data Interface (PDI)	≥ 55 ms

8.1.3 PROFIBUS DP

Tab. 152: Technical specifications FX0-GPRO

Interface	Minimum	Typical	Maximum
Fieldbus	PROFIBUS DP-V0		
Interface level	RS-485		
Connector technology	D-Sub socket 9-pin		
Slave address (set via rotary switches)	0		99
Slave address (set via Flexi Soft Designer)[18)]	3		125
Baud rate (automatic adjustment)			12 MBaud
Baud rate (kbit/s with standard cable) 9.6/19.2/93.75 187.5 500 1500 12000			Max. cable length 1200 m 1000 m 400 m 200 m 100 m

Cable parameters see section 6.1 “PROFIBUS DP gateway” from page 116.

8.1.4 CANopen

Tab. 153: Technical specifications FX0-GCAN

Interface	Minimum	Typical	Maximum
Fieldbus	CANopen DS-301		
Interface level	RS-485		
Connector technology	5 pin open style connector		
Slave address (set via rotary switches)	0		99
Slave address (set via Flexi Soft Designer)[19)]	1		127
Baud rate (kbit/s with standard cable) 125 250 500 800 1000			Max. cable length 500 m 250 m 100 m 40 m 20 m

Cable parameters see chapter 6.2 “CANopen gateway” from page 130.

[18)] In order to set the slave address via software, the hardware setting for the address must be “00”.
[19)] In order to set the slave address via software, the hardware setting for the address must be “00”.

8.1.5 DeviceNet

Tab. 154: Technical specifications FX0-GDEV

Interface	Minimum	Typical	Maximum
Fieldbus	DeviceNet		
Interface level	RS-485		
Connector technology	5-pin open style connector		
Slave address (via rotary switch)	0		63
Slave address (via Flexi Soft Designer)[20]	1		63
Baud rate (kbits/s with standard cable) 125 250 500			Max. cable length 500 m 250 m 100 m

Cable parameters see section 6.3 “DeviceNet gateway” on page 160.

8.2 Technical specifications, supply circuit

These technical specifications apply to all gateways.

Tab. 155: Technical specifications supply circuit

Supply circuit (e.g. via FLEXBUS+)	Minimum	Typical	Maximum
Supply voltage	16.8 V DC	24 V DC	30 V DC
Power consumption			
FX0-GPRO, FX0-GCAN, FX0-GDEV	–	–	1.6 W
FX0-GENT, FX0-GMOD, FX0-GPNT	–	–	2.4 W
FX0-GETC	–	–	3 W

[20] In order to set the slave address via software, the hardware setting for the address must be “00”.

8.3 General technical specifications

These technical specifications apply to all gateways.

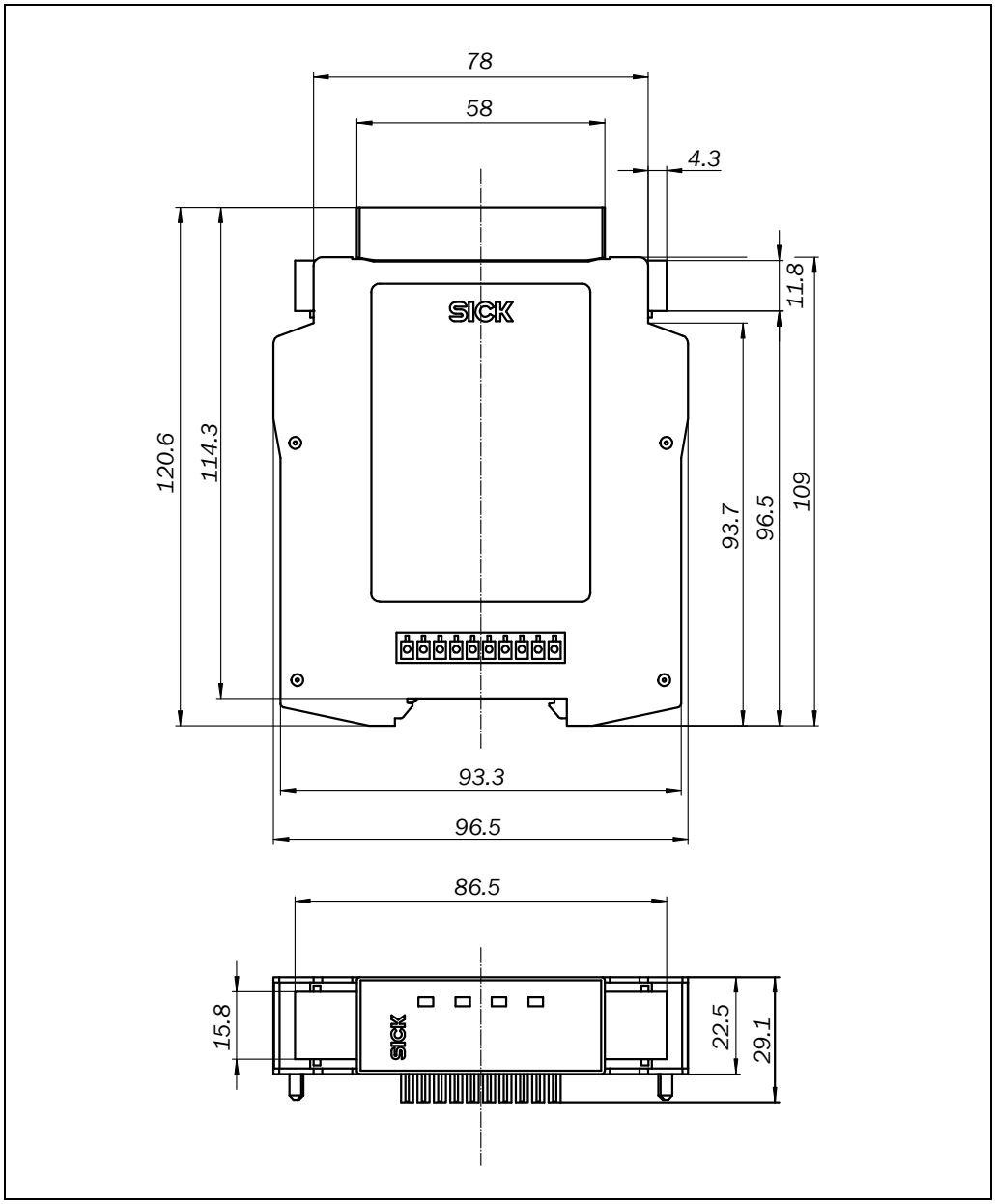
Tab. 156: General technical specifications

Terminals	
Fieldbus	See section 8.1 “Technical specifications gateways”
FLEXBUS+	10-pin connector for internal safety bus (plug)
Climatic conditions	
Ambient operating temperature T_A	–25 to +55 °C
Storage temperature	–25 to +70 °C
Relative humidity	10 % to 95 %, non-condensing
Climatic conditions	
Air pressure in operation	860 to 1060 hPa (EN 61131-2)
Mechanical strength	
Vibration resistance	10-500 Hz/5g (EN 60068-2-6)
Shock resistance	
Continuous shock	10 g, 16 ms (EN 60068-2-29)
Single shock	30 g, 11 ms (EN 60068-2-27)
Electrical safety (see FX3-CPUx)	
Enclosure rating	IP 20 (EN 60529)
Protection class	III (EN 61140)
Electromagnetic compatibility	Class A (EN 61000-6-2/EN 55011)
Mechanical and assembly	
Housing	
Material	Polycarbonate
Type	Device for control cabinet installation
Enclosure rating	
Housing	IP 20 (EN 60529)
Terminals	IP 40 (EN 60529)
Colour	
Gateways	Light grey
Weight	
FX0-GENT, FX0-GMOD, FX0-GPNT	125 g (± 10 %)
FX0-GPRO, FX0-GETC, FX0-GCAN, FX0-GDEV	150 g (± 10 %)
FLEXBUS+ connection (internal bus)	
Number of poles	10
Connectors	1 plug left, 1 socket right
Mounting rail	DIN mounting rail acc. to EN 60715

8.4 Dimensional drawings

8.4.1 Dimensional drawing FX0-GENT, FX0-GMOD, FX0-GPNT and FX0-GETC

Fig. 91: Dimensional drawing FX0-GENT, FX0-GMOD, FX0-GPNT and FX0-GETC (mm)



8.4.2 Dimensional drawing FX0-GPRO, FX0-GCAN and FX0-GDEV

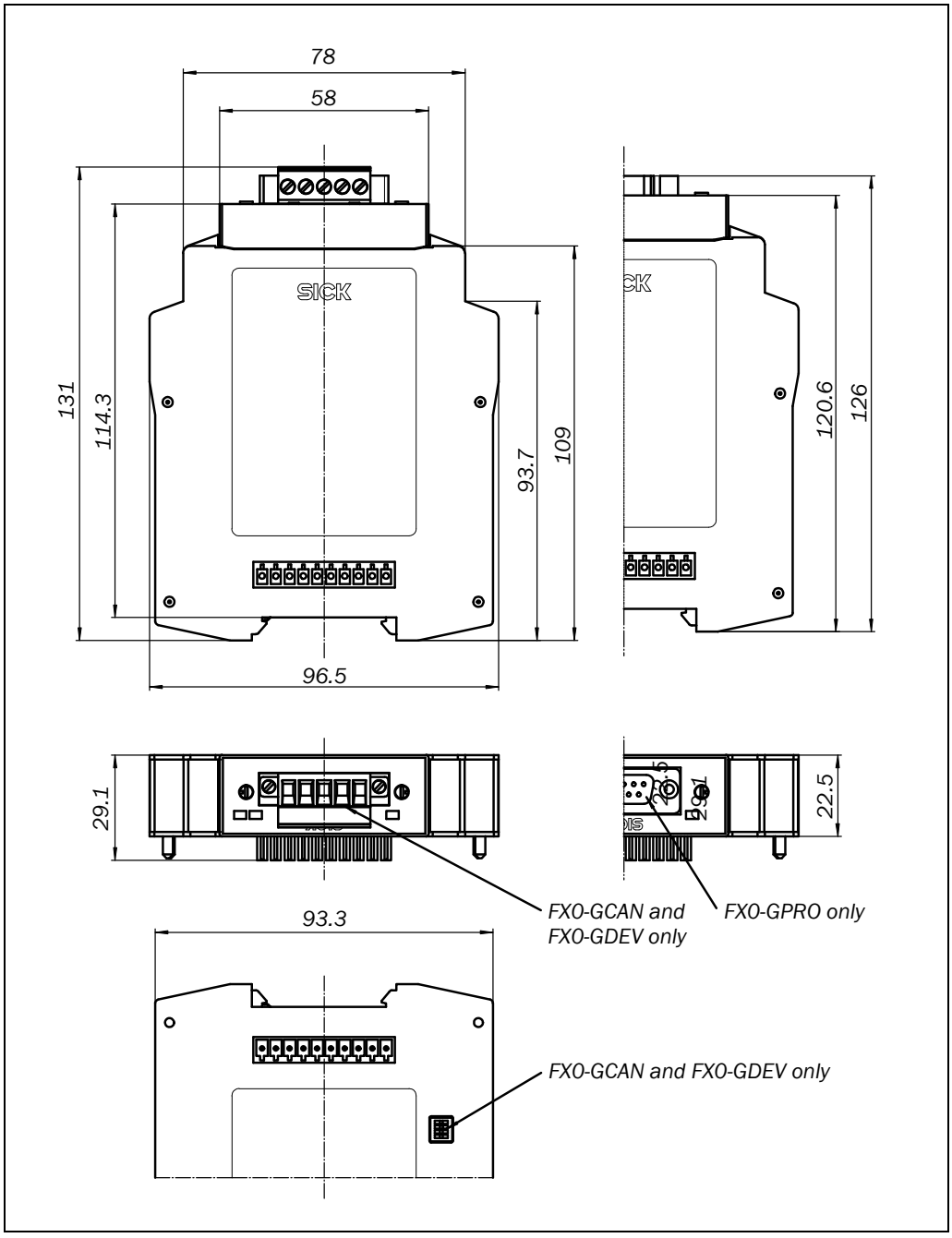


Fig. 92: Dimensional drawing FX0-GPRO, FX0-GCAN and FX0-GDEV (mm)

8.5 Ordering information Flexi Soft gateways

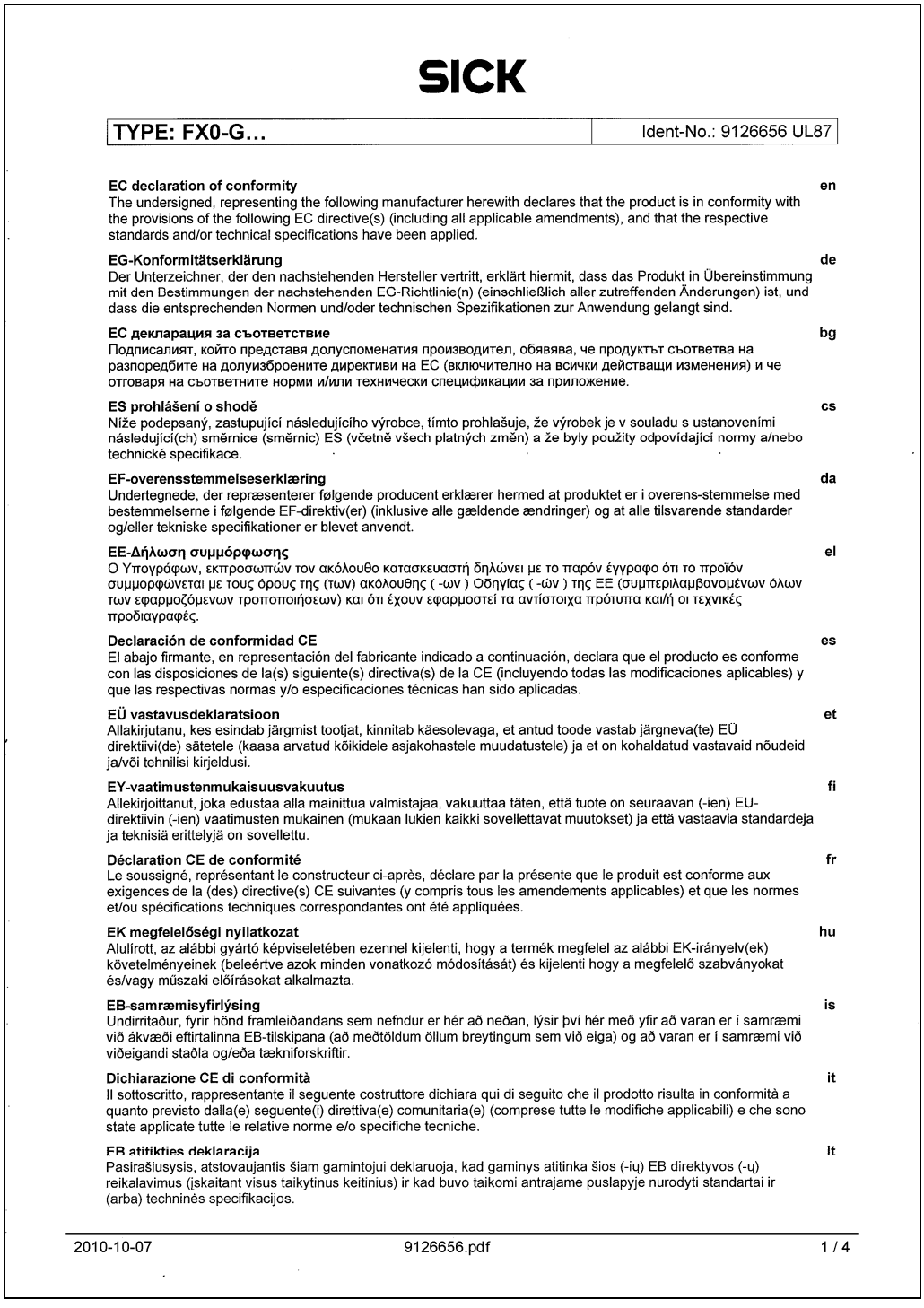
Tab. 157: Ordering information Flexi Soft gateways

Type	Gateway	Part number
FX0-GENT	EtherNet/IP	1044072
FX0-GMOD	Modbus TCP	1044073
FX0-GPNT	PROFINET IO	1044074
FX0-GPRO	PROFIBUS DP	1044075
FX0-GCAN	CANopen	1044076
FX0-GDEV	DeviceNet	1044077
FX0-GETC	EtherCAT	1051432

9 Annex

9.1 EC declaration of conformity

Fig. 93: EC declaration of conformity (page 1)



SICK

TYPE: FX0-G...	Ident-No.: 9126656 UL87

EC declaration of conformity en
The undersigned, representing the following manufacturer herewith declares that the product is in conformity with the provisions of the following EC directive(s) (including all applicable amendments), and that the respective standards and/or technical specifications have been applied.

EG-Konformitätserklärung de
Der Unterzeichner, der den nachstehenden Hersteller vertritt, erklärt hiermit, dass das Produkt in Übereinstimmung mit den Bestimmungen der nachstehenden EG-Richtlinie(n) (einschließlich aller zutreffenden Änderungen) ist, und dass die entsprechenden Normen und/oder technischen Spezifikationen zur Anwendung gelangt sind.

ЕС декларация за съответствие bg
Подписалият, който представя долуспоменатия производител, обявява, че продуктът съответва на разпоредбите на долуизброените директиви на ЕС (включително на всички действащи изменения) и че отговаря на съответните норми и/или технически спецификации за приложение.

ES prohlášení o shodě cs
Níže podepsaný, zastupující následujícího výrobce, tímto prohlašuje, že výrobek je v souladu s ustanoveními následující(ch) směrnice (směrnic) ES (včetně všech platných změn) a že byly použity odpovídající normy a/nebo technické specifikace.

EF-overensstemmelseserklæring da
Undertegnede, der repræsenterer følgende producent erklærer hermed at produktet er i overens-stemmelse med bestemmelserne i følgende EF-direktiv(er) (inklusive alle gældende ændringer) og at alle tilsvarende standarder og/eller tekniske specifikationer er blevet anvendt.

ΕΕ-Δήλωση συμμόρφωσης el
Ο Υπογράφων, εκπροσωπών τον ακόλουθο κατασκευαστή δηλώνει με το παρόν έγγραφο ότι το προϊόν συμμορφώνεται με τους όρους της (των) ακόλουθης (-ων) Οδηγίας (-ών) της ΕΕ (συμπεριλαμβανομένων όλων των εφαρμοζόμενων τροποποιήσεων) και ότι έχουν εφαρμοστεί τα αντίστοιχα πρότυπα και/ή οι τεχνικές προδιαγραφές.

Declaración de conformidad CE es
El abajo firmante, en representación del fabricante indicado a continuación, declara que el producto es conforme con las disposiciones de la(s) siguiente(s) directiva(s) de la CE (incluyendo todas las modificaciones aplicables) y que las respectivas normas y/o especificaciones técnicas han sido aplicadas.

EÜ vastavusdeklaratsioon et
Allakirjutanu, kes esindab järgmist tootjat, kinnitab käesolevaga, et antud toode vastab järgneva(te) EÜ direktiivi(de) sätetele (kaasa arvatud kõikidele asjakohastele muudatustele) ja et on kohaldatud vastavaid nõudeid ja/või tehnilisi kirjeldusi.

EY-vaatimustenmukaisuusvakuutus fi
Allekirjoittanut, joka edustaa alla mainittua valmistajaa, vakuuttaa täten, että tuote on seuraavan (-ien) EU-direktiivin (-ien) vaatimusten mukainen (mukaan lukien kaikki sovellettavat muutokset) ja että vastaavia standardeja ja teknisiä erittelyjä on sovellettu.

Déclaration CE de conformité fr
Le soussigné, représentant le constructeur ci-après, déclare par la présente que le produit est conforme aux exigences de la (des) directive(s) CE suivantes (y compris tous les amendements applicables) et que les normes et/ou spécifications techniques correspondantes ont été appliquées.

EK megfelelőségi nyilatkozat hu
Alulírott, az alábbi gyártó képviseletében ezennel kijelenti, hogy a termék megfelel az alábbi EK-irányelv(ek) követelményeinek (beleértve azok minden vonatkozó módosítását) és kijelenti hogy a megfelelő szabványokat és/vagy műszaki előírásokat alkalmazta.

EB-samræmisyfirlýsing is
Undirritaður, fyrir hönd framleiðandans sem nefndur er hér að neðan, lýsir því hér með yfir að varan er í samræmi við ákvæði eftirtalinna EB-tilskipana (að meðtöldum öllum breytingum sem við eiga) og að varan er í samræmi við viðeigandi staðla og/eða tækniforskriftir.

Dichiarazione CE di conformità it
Il sottoscritto, rappresentante il seguente costruttore dichiara qui di seguito che il prodotto risulta in conformità a quanto previsto dalla(e) seguente(i) direttiva(e) comunitaria(e) (comprese tutte le modifiche applicabili) e che sono state applicate tutte le relative norme e/o specifiche tecniche.

EB atitikties deklaracija lt
Pasirašiusysis, atstovaujantis šiam gamintojui deklaruoja, kad gaminys atitinka šios (-ių) EB direktyvos (-ų) reikalavimus (įskaitant visus taikytinus keitinius) ir kad buvo taikomi antrajame puslapyje nurodyti standartai ir (arba) techninės specifikacijos.

2010-10-07 9126656.pdf 1 / 4

Fig. 94: EC declaration of conformity (page 2)

SICK

TYPE: FX0-G…	Ident-No.: 9126656 UL87

EK atbilstības deklarācija lv
Apakšā parakstījusies persona, kas pārstāv zemāk minēto ražotāju ar šo deklarē, ka izstrādājums atbilst zemāk minētajai (-ām) EK direktīvai (-ām) (ieskaitot visus atbilstošos grozījumus) un ka izstrādājumam ir piemēroti attiecīgie standarti un/vai tehniskās specifikācijas.

EG-verklaring van overeenstemming nl
Ondergetekende, vertegenwoordiger van de volgende fabrikant, verklaart hiermee dat het product voldoet aan de bepalingen van de volgende EG-richtlijn(en) (inclusief alle van toepassing zijnde wijzigingen) en dat de overeenkomstige normen en/of technische specificaties zijn toegepast.

EF-samsvarserklæring no
Undertegnede, som representerer nedennevnte produsent, erklærer herved at produktet er i samsvar med bestemmelsene i følgende EU-direktiv(er) (inkludert alle relevante endringer) og at relevante normer og/eller tekniske spesifikasjoner er blitt anvendt.

Deklaracja zgodności WE pl
Niżej podpisany, reprezentujący następującego producenta niniejszym oświadcza, że wyrób jest zgodny z postanowieniami następujących dyrektyw WE (wraz z odnośnymi poprawkami) oraz, że zastosowano odpowiednie normy i/lub specyfikacje techniczne.

Declaração CE de conformidade pt
O abaixo assinado, que representa o seguinte fabricante, declara deste modo que o produto está em conformidade com as disposições da(s) seguinte(s) directiva(s) CE (incluindo todas as alterações aplicáveis) e que foram aplicadas as respectivas normas e/ou especificações técnicas.

Declarație de conformitate CE ro
Semnatarul, în calitate de reprezentant al producătorului numit mai jos, declară prin prezenta că produsul este în conformitate cu prevederile directivelor CE enumerate mai jos (inclusiv cu toate modificările aferente) şi că s-au întrunit normele şi/sau specificaţiile tehnice corespunzătoare.

ES vyhlásenie o zhode sk
Dolu podpísaný zástupca výrobcu týmto vyhlasuje, že výrobok je v súlade s ustanoveniami nasledujúcej (nasledujúcich) smernice (smerníc) ES (vrátane všetkých platných zmien) a že sa použili príslušné normy a/alebo technické špecifikácie.

Izjava ES o skladnosti sl
Podpisani predstavnik spodaj navedenega proizvajalca izjavljam, da je proizvod v skladu z določbami spodaj navedenih direktiv ES (vključno z vsemi ustreznimi spremembami) in da so bili uporabljeni ustrezni standardi in/ali tehnične specifikacije.

EG-försäkran om överensstämmelse sv
Undertecknad, som representerar nedanstående tillverkare, försäkrar härmed att produkten överensstämmer med bestämmelserna i följande EU-direktiv (inklusive samtliga tillämpliga tillägg till dessa) och att relevanta standarder och/eller tekniska specifikationer har tillämpats.

AB-Uygunluk Beyanı tr
Aşağıdaki üreticiyi temsil eden imza sahibi böylelikle, ürünün aşağıdaki AB-Yönergesinin(lerin) direktifleri ile (tüm ilgili değişiklikleri kapsayacak şekilde) uyumlu olduğunu ve ilgili normların ve/veya teknik spesifikasyonların uygulandığını beyan eder.

Directives used:	EMC-DIRECTIVE	2004/108/EC
	EMC;IMMUNITY FOR INDUSTRIAL ENVIRONMENTS	EN 61000- 6- 2
	INDUSTRIAL, SCIENTIFIC AND MEDICAL (ISM)	EN 55011

Product: **FX0-G…**

You can obtain the EC declaration of conformity with the standards used at: **www.sick.com**, search: 9126656

SICK AG
Erwin-Sick-Straße 1
D-79183 Waldkirch
Germany

2010-09-30
Date

ppa. Dr.Georg Plasberg
Management Board
(Industrial Safety Systems)
authorized for technical documentation

ppa. Birgit Knobloch
Division Manager Production
(Industrial Safety Systems)

2010-10-07 9126656.pdf 2 / 4

9.2 List of tables

Tab. 1: Overview on disposal by components 11
Tab. 2: Device variants and features overview 13
Tab. 3: Firmware versions of the Ethernet gateways 13
Tab. 4: Availability of data set 1-4 15
Tab. 5: Overview input data sets 1-3 (default settings for EtherNet/IP, Modbus TCP and TCP/IP) 16
Tab. 6: Module status 17
Tab. 7: Usage of the more significant bits on dual-channel evaluation on I/O modules FX3-XTIO 18
Tab. 8: Module status bits of the FX3-CPUx main modules 19
Tab. 9: Module status bits of the I/O modules FX3-XTIO and FX3-XTDI 20
Tab. 10: Module status bits of the I/O modules FX3-XTDS 20
Tab. 11: Module status bits of the I/O modules FX0-STIO 20
Tab. 12: Module status bits of the motion control modules FX3-MOCx 21
Tab. 13: Module status bits of the gateways 21
Tab. 14: Example module status in data set 3 21
Tab. 15: Example module status byte 0 of module 2 21
Tab. 16: Example module status byte 3 of module 2 22
Tab. 17: Output data blocks 1-5 for the different gateways 23
Tab. 18: Number of possible TCP/IP connections 37
Tab. 19: Telegram structure 40
Tab. 20: Error response message 40
Tab. 21: Get data set(s) request 41
Tab. 22: Get data set(s) response 41
Tab. 23: Write output data set setting command 42
Tab. 24: Write output data set setting response 42
Tab. 25: Auto update mode configuration command 44
Tab. 26: Auto update mode configuration response 44
Tab. 27: Auto update mode normal operation message 46
Tab. 28: TCP/IP process image example 46
Tab. 29: Meaning of the LED displays of the FX0-GENT 53
Tab. 30: Examples for the packet update interval 56
Tab. 31: Recommended bandwidths for Class 1 messages 56
Tab. 32: Class 1 read access points to input data sets 57
Tab. 33: Class 1 write access points to output data sets 58
Tab. 34: Assembly object class attributes 58
Tab. 35: Assembly object instance definitions 59
Tab. 36: Assembly object instance attributes 60
Tab. 37: Assembly object common services 60
Tab. 38: Number of possible connections 61
Tab. 39: Configuration guideline — Gateway as master 61
Tab. 40: Configuration guideline — Gateway as slave 61

Tab. 41: Full Data Set Transfer object (72h) class attribute (instance 0) ... 65
Tab. 42: Full Data Set Transfer object (72h) instance attribute (instance 1) ... 65
Tab. 43: Full Data Set Transfer object (72h) common services ... 66
Tab. 44: Individual Input Data Set Transfer object (73h) class attributes ... 66
Tab. 45: Individual Input Data Set Transfer object (73h) instance attributes ... 66
Tab. 46: Individual Input Data Set Transfer object (73h) common services ... 66
Tab. 47: Individual Input Data Set Transfer object (73h) instance 1 attribute definitions ... 67
Tab. 48: Individual Input Data Set Transfer object (73h) instance 2 attribute definitions ... 67
Tab. 49: Individual Input Data Set Transfer object (73h) instance 3 attribute definitions ... 67
Tab. 50: Individual Input Data Set Transfer object (73h) instance 4 attribute definitions ... 67
Tab. 51: Supported PCCC messages for the PLC-5, SLC and MicroLogix PLCs ... 69
Tab. 52: Addressing for the PLC-5/SLC messages ... 70
Tab. 53: PCCC object (67h) common services ... 70
Tab. 54: PCCC object (67h) request message ... 71
Tab. 55: PCCC object (67h) response message ... 71
Tab. 56: PCCC object (67h) supported PCCC command types ... 71
Tab. 57: Troubleshooting for the FX0-GENT ... 74
Tab. 58: Meaning of the LED displays ... 75
Tab. 59: Number of possible connections ... 77
Tab. 60: Configuration guideline — Gateway as master ... 78
Tab. 61: Configuration guideline — Gateway as slave ... 79
Tab. 62: Data addressing for FX0-GMOD as receiver ... 82
Tab. 63: Modbus commands ... 83
Tab. 64: Modbus error messages ... 83
Tab. 65: Troubleshooting for the FX0-GMOD ... 83
Tab. 66: Meaning of the LED displays of the FX0-GPNT ... 85
Tab. 67: Default content of input data block 1-5 of the FX0-GPNT ... 92
Tab. 68: Memory location for data set 2, 3 and 4 ... 93
Tab. 69: Default content of input data set 2-4 of the FX0-GPNT. ... 94
Tab. 70: I&M information of the FX0-GPNT ... 95
Tab. 71: PROFINET IO error type definitions ... 96
Tab. 72: Troubleshooting for the FX0-GPNT ... 100
Tab. 73: Meaning of the status LEDs of the FX0-GETC ... 102
Tab. 74: Process data objects of the FX0-GETC ... 106
Tab. 75: Default content of input data set 1-5 of the FX0-GETC ... 107
Tab. 76: FX0-GETC Input Process Data CoE object 2000h ... 112
Tab. 77: FX0-GETC Output Process Data CoE object 2001h ... 112
Tab. 78: FX0-GETC CRC CoE object 2002h ... 112
Tab. 79: FX0-GETC Status and Diagnosis CoE object 2003h ... 112

Tab. 80: Structure of the Diagnosis history object ... 113
Tab. 81: Module status bits of the FX0-GETC ... 114
Tab. 82: Troubleshooting for the FX0-GETC ... 115
Tab. 83: Meaning of the status LEDs of the FX0-GPRO ... 116
Tab. 84: Address switch FX0-GPRO ... 117
Tab. 85: Explanation bus cable FX0-GPRO ... 118
Tab. 86: Cable parameters FX0-GPRO ... 118
Tab. 87: Maximum cable lengths FX0-GPRO ... 119
Tab. 88: GSD file versions for the FX0-GPRO ... 119
Tab. 89: Default content of input data block 1-5 of the FX0-GPRO ... 121
Tab. 90: Content of the PROFIBUS diagnostics messages ... 124
Tab. 91: Flexi Soft module numbers ... 124
Tab. 92: PROFIBUS error messages ... 125
Tab. 93: Troubleshooting for the FX0-GPRO ... 129
Tab. 94: Meaning of the status LEDs of the FX0-GCAN ... 130
Tab. 95: Address switch FX0-GCAN ... 131
Tab. 96: DIP switch settings on the FX0-GCAN ... 131
Tab. 97: Maximum cable lengths FX0-GCAN ... 133
Tab. 98: CAN identifier structure ... 136
Tab. 99: PCS communication objects ... 136
Tab. 100: Network management for one NMT slave with address N ... 137
Tab. 101: Network management for all NMT slaves ... 137
Tab. 102: Example NMT object for resetting all communication ... 137
Tab. 103: Polling inputs using SYNC ... 138
Tab. 104: Emergency messages ... 138
Tab. 105: CANopen emergency messages ... 139
Tab. 106: CANopen emergency, diagnostic bits M2 to M5 ... 141
Tab. 107: Request from NMT master ... 142
Tab. 108: Response from slave ... 142
Tab. 109: Remote transmission request ... 142
Tab. 110: Default content of the transmit process data objects (TxPDOs) of the FX0-GCAN ... 143
Tab. 111: TxPDO 1...4 ... 144
Tab. 112: RxPDO 1...4 ... 144
Tab. 113: Write SDO ... 145
Tab. 114: Write SDO confirmation ... 145
Tab. 115: Read SDO ... 145
Tab. 116: Read SDO confirmation ... 145
Tab. 117: Example — Read SDO ... 146
Tab. 118: Example — Read SDO confirmation ... 146
Tab. 119: Supported SDOs ... 146
Tab. 120: SDO 1018 contents ... 148
Tab. 121: SDO 1027 contents ... 148

Tab. 122: SDO 1400...1403 contents 149
Tab. 123: SDO 1800...1803 contents 149
Tab. 124: TxPDO transmission types 149
Tab. 125: SDO 3100 contents 150
Tab. 126: Module type codes in SDO 3300 151
Tab. 127: Mapping table for SDO 6000 — RxPDO 1-4 152
Tab. 128: Mapping table for SDO 6200 — TxPDO 1-4 153
Tab. 129: Guarding protocol overview and comparison 154
Tab. 130: Emergency states and transitions 155
Tab. 131: Error objects overview 156
Tab. 132: Troubleshooting for the FX0-GCAN 159
Tab. 133: Meaning of the status LEDs of the FX0-GDEV 161
Tab. 134: Address switches on the FX0-GDEV 162
Tab. 135: DIP switch settings on the FX0-GDEV 162
Tab. 136: Maximum cable lengths FX0-GDEV 164
Tab. 137: Assembly object 167
Tab. 138: Produced Assembly Instance 167
Tab. 139: Consumed Assembly Instance 168
Tab. 140: Class attributes for the Individual Data Set Transfer object (73h) 168
Tab. 141: Instance attributes for the Individual Data Set Transfer object (73h) 168
Tab. 142: Individual Input Data Set Transfer object (73h) general services 168
Tab. 143: Individual Input Data Set Transfer object (73h) instance 1 attribute definitions 169
Tab. 144: Individual Input Data Set Transfer object (73h) instance 2 attribute definitions 169
Tab. 145: Individual Input Data Set Transfer object (73h) instance 3 attribute definitions 169
Tab. 146: Individual Input Data Set Transfer object (73h) instance 4 attribute definitions 170
Tab. 147: Troubleshooting for the FX0-GDEV 171
Tab. 148: Content of the data sets 1-4 173
Tab. 149: Default configuration for the operational data transmitted into the network 174
Tab. 150: Technical specifications FX0-GENT, FX0-GMOD and FX0-GPNT 183
Tab. 151: Technical specifications FX0-GETC 183
Tab. 152: Technical specifications FX0-GPRO 184
Tab. 153: Technical specifications FX0-GCAN 184
Tab. 154: Technical specifications FX0-GDEV 185
Tab. 155: Technical specifications supply circuit 185
Tab. 156: General technical specifications 186
Tab. 157: Ordering information Flexi Soft gateways 188

9.3 List of illustrations

Fig. 1: Mounting the module onto the DIN mounting rail 25
Fig. 2: Installing the end clips 25
Fig. 3: Removing the removable terminals 26
Fig. 4: Disconnecting the plug connections 26
Fig. 5: Removing modules from the DIN mounting rail 26
Fig. 6: Connection settings dialog 28
Fig. 7: Modify profile dialog 28
Fig. 8: Hardware configuration dialog 29
Fig. 9: Ethernet gateway configuration dialog 33
Fig. 10: Connection settings dialog 34
Fig. 11: Create new TCP/IP profile dialog 34
Fig. 12: Create new TCP/IP profile dialog after scan has been performed 35
Fig. 13: Connection settings dialog with new TCP/IP profile 36
Fig. 14: Connection settings dialog with new TCP/IP profile activated 36
Fig. 15: Found gateways in the Network scan dialog 37
Fig. 16: TCP/IP configuration dialog 38
Fig. 17: TCP/IP configuration for Application requests (Polling) mode 43
Fig. 18: TCP/IP configuration for auto update 45
Fig. 19: Socket monitor window 48
Fig. 20: PC is TCP Client area — Gateway is configured to listen 49
Fig. 21: PC is TCP Server area — Gateway is configured to connect 49
Fig. 22: Write data to device — Read input data sets 50
Fig. 23: Write data to device — Write output data sets 50
Fig. 24: Write data to device — Configure input data sets 51
Fig. 25: Read automatic data from device 52
Fig. 26: Interfaces and display elements of the FX0-GENT 53
Fig. 27: EtherNet/IP gateway configuration dialog 54
Fig. 28: EtherNet/IP gateway configuration as master 62
Fig. 29: Example of tag names in a PLC program 63
Fig. 30: EtherNet/IP gateway configuration as slave 64
Fig. 31: Creating tags for explicit messaging 72
Fig. 32: Main routine programming for explicit messaging 72
Fig. 33: Explicit messaging — Message configuration 73
Fig. 34: Explicit messaging — Communication configuration 73
Fig. 35: Interfaces and display elements of the FX0-GMOD 75
Fig. 36: Modbus TCP gateway configuration dialog 76
Fig. 37: Modbus TCP gateway configuration as master 79
Fig. 38: Modbus TCP gateway configuration as slave 81
Fig. 39: Interfaces and display elements of the FX0-GPNT 85
Fig. 40: Disabling the STATUS LED of the FX0-GPNT 86
Fig. 41: PROFINET IO gateway configuration dialog 87

Fig. 42: PROFINET IO gateway in the PROFINET IO HW Config 89
Fig. 43: Configuration of the update time of the FX0-GPNT 90
Fig. 44: Properties dialog of the FX0-GPNT 91
Fig. 45: Assign device name dialog for the FX0-GPNT 91
Fig. 46: Projecting the FX0-GPNT 93
Fig. 47: Controls and status indicators FX0-GETC 101
Fig. 48: Gateway configuration dialog for the FX0-GETC 104
Fig. 49: Example for inserting the FX0-GETC into an EtherCAT network 105
Fig. 50: PDO configuration in the EtherCAT network configuration tool 106
Fig. 51: EtherCAT gateway configuration dialog 109
Fig. 52: Edit the data block start address 109
Fig. 53: EoE activation for the FX0-GETC in TwinCAT 110
Fig. 54: Enabling TCP/IP routing for the FX0-GETC in the Flexi Soft Designer 111
Fig. 55: CoE object catalogue of the FX0-GETC in TwinCAT 111
Fig. 56: Controls and status indicators FX0-GPRO 116
Fig. 57: Setting the PROFIBUS address for the FX0-GPRO 117
Fig. 58: D-Sub socket and plug pin assignments FX0-GPRO 118
Fig. 59: Bus cable FX0-GPRO 118
Fig. 60: PROFIBUS DP configuration example in Siemens SIMATIC Manager 120
Fig. 61: PROFIBUS DP gateway configuration dialog 122
Fig. 62: Edit the data block start address 122
Fig. 63: PROFIBUS DP gateway in the PROFIBUS HW Config 123
Fig. 64: Controls and status indicators FX0-GCAN 130
Fig. 65: DIP switch settings on the FX0-GCAN 131
Fig. 66: Setting the CANopen address for the FX0-GCAN 132
Fig. 67: Open style connector and assignment FX0-GCAN 132
Fig. 68: CANopen bus cable 133
Fig. 69: CoDeSys PLC Configuration editor window 134
Fig. 70: Appending a CanMaster using CoDeSys 2.3 134
Fig. 71: Appending the FX0-GCAN using CoDeSys 2.3 135
Fig. 72: PDO configuration using CoDeSys 2.3 135
Fig. 73: PDO properties dialog in CoDeSys 2.3 135
Fig. 74: Controls and status indicators FX0-GDEV 161
Fig. 75: DIP switch settings on the FX0-GDEV 162
Fig. 76: Setting the DeviceNet address of the FX0-GDEV 163
Fig. 77: Open style connector and assignment FX0-GDEV 163
Fig. 78: DeviceNet bus cable 164
Fig. 79: Configuration of the type of communication based on the example of the DeviceNetManagerTM from Allen Bradley 165
Fig. 80: Selection of input data set 1 and output data set 1 using DeviceNetManagerTM from Allen Bradley 165
Fig. 81: Selection of input data set 3 and output data set 1-5 using DeviceNetManagerTM from Allen Bradley 166

Fig. 82: Default configuration for the operational data transmitted into the network 175
Fig. 83: Toolbar for the routing configuration 176
Fig. 84: Gateway direct output values in the default process image 178
Fig. 85: Defining additional gateway direct output values in the process image 179
Fig. 86: Network to Flexi Soft dialog of the FX0-GMOD 180
Fig. 87: Tag names of incoming bits in the logic editor dialog of the FX3-CPUx 181
Fig. 88: Active and inactive bits in the online process image 182
Fig. 89: Inactive Network input bits as a result of an error. 182
Fig. 90: Network output bits with no tag names assigned in the online process image 182
Fig. 91: Dimensional drawing FX0-GENT, FX0-GMOD, FX0-GPNT and FX0-GETC (mm) 187
Fig. 92: Dimensional drawing FX0-GPRO, FX0-GCAN and FX0-GDEV (mm) 188
Fig. 93: EC declaration of conformity (page 1) 189
Fig. 94: EC declaration of conformity (page 2) 190

8012664/XB29/2013-06-11 · RV/XX (2013-06) · A4 sw int40

Australia
Phone +61 3 9457 0600
1800 334 802 – tollfree
E-Mail sales@sick.com.au

Belgium/Luxembourg
Phone +32 (0)2 466 55 66
E-Mail info@sick.be

Brasil
Phone +55 11 3215-4900
E-Mail sac@sick.com.br

Canada
Phone +1 905 771 14 44
E-Mail information@sick.com

Ceská Republika
Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

China
Phone +86 4000 121 000
E-Mail info.china@sick.net.cn
Phone +852-2153 6300
E-Mail ghk@sick.com.hk

Danmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Deutschland
Phone +49 211 5301-301
E-Mail kundenservice@sick.de

España
Phone +34 93 480 31 00
E-Mail info@sick.es

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Great Britain
Phone +44 (0)1727 831121
E-Mail info@sick.co.uk

India
Phone +91–22–4033 8333
E-Mail info@sick-india.com

Israel
Phone +972-4-6881000
E-Mail info@sick-sensors.com

Italia
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 (0)3 3358 1341
E-Mail support@sick.jp

Magyarország
Phone +36 1 371 2680
E-Mail office@sick.hu

Nederlands
Phone +31 (0)30 229 25 44
E-Mail info@sick.nl

Norge
Phone +47 67 81 50 00
E-Mail austefjord@sick.no

Österreich
Phone +43 (0)22 36 62 28 8-0
E-Mail office@sick.at

Polska
Phone +48 22 837 40 50
E-Mail info@sick.pl

România
Phone +40 356 171 120
E-Mail office@sick.ro

Russia
Phone +7-495-775-05-30
E-Mail info@sick.ru

Schweiz
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Singapore
Phone +65 6744 3732
E-Mail admin@sicksgp.com.sg

Slovenija
Phone +386 (0)1-47 69 990
E-Mail office@sick.si

South Africa
Phone +27 11 472 3733
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321/4
E-Mail info@sickkorea.net

Suomi
Phone +358-9-25 15 800
E-Mail sick@sick.fi

Sverige
Phone +46 10 110 10 00
E-Mail info@sick.se

Taiwan
Phone +886-2-2375-6288
E-Mail sales@sick.com.tw

Türkiye
Phone +90 (216) 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 (0) 4 8865 878
E-Mail info@sick.ae

USA/México
Phone +1(952) 941-6780
1 800 325-7425 – tollfree
E-Mail info@sickusa.com

More representatives and agencies
at **www.sick.com**

SICK
Sensor Intelligence.