OPERATING INSTRUCTIONS

Flexi Soft Gateways in Flexi Soft Designer

Configuration software

SICK
Sensor Intelligence.

Described product

Flexi Soft Gateways in Flexi Soft Designer

Configuration Software

Manufacturer

SICK AG
Erwin-Sick-Str. 1
79183 Waldkirch
Germany

Legal information

Original document

This document is an original document of SICK AG.

Contents

1 About this document

1.1 Purpose of this document

These operating instructions contain the information needed during the life cycle of the Flexi Soft gateways.

These operating instructions must be made available to all people who work with the Flexi Soft gateways and the Flexi Soft Designer configuration software.

For the Flexi Soft system, there are operating instructions and mounting instructions, each covering clearly defined fields of application.

Table 1: Overview of the Flexi Soft documentation

Document type	Title	Contents	Purpose	Part number
Operating instructions	Flexi Soft Modular Safety Controller Hardware	Description of the Flexi Soft modules and their functions	Instructions for technical personnel working for the machine manufacturer or operator on the safe mounting, electrical installation, and maintenance of the Flexi Soft safety controller	8012999
Operating instructions	Flexi Soft in the Flexi Soft Designer Configuration software	Description of the software-based configuration of the Flexi Soft safety controller along with important diagnostics functions and detailed notes on identifying and rectifying errors	Instructions for technical personnel working for the machine manufacturer or operator on the safe configuration and commissioning, as well as the safe operation, of the Flexi Soft safety controller	8012998
Operating instructions	Safety Designer Configuration software	Description of the installation and general basic principles of operation	To provide technical personnel working for the machine manufacturer/operator with instructions so that they can use the Safety Designer configuration software	8018178
Operating instructions	Flexi Soft in the Safety Designer Configuration software	Description of the software-based configuration of the Flexi Soft safety controller along with important diagnostics functions and detailed notes on identifying and rectifying errors	Instructions for technical personnel working for the machine manufacturer or operator on the safe configuration and commissioning, as well as the safe operation, of the Flexi Soft safety controller	8013926
Operating instructions	Flexi Soft Gateways Hardware	Description of the Flexi Soft gateways and their functions	To provide technical personnel working for the machine manufacturer/operator with instructions so that they can safely carry out the mounting, electrical installation, and maintenance work for the Flexi Soft gateways	8012662
Operating instructions	Flexi Soft Gateways in Flexi Soft Designer Configuration software	Description of the software-based configuration of the Flexi Soft gateway, information about data exchange in networks as well as about the status, planning, and associated mapping	To provide technical personnel working for the machine manufacturer/operator with instructions so that they can safely configure and commission the Flexi Soft gateways	8012483

Document type	Title	Contents	Purpose	Part number
Operating instructions	Flexi Soft Gateways in the Safety Designer Configuration software	Description of the software-based configuration of the Flexi Soft gateway, information about data exchange in networks as well as about the status, planning, and associated mapping	To provide technical personnel working for the machine manufacturer/operator with instructions so that they can safely configure and commission the Flexi Soft gateways	8018170
Operating instructions	Flexi Loop safe series connection Hardware	Description of the Flexi Loop safe series connection and its functions	To provide technical personnel working for the machine manufacturer/operator with instructions so that they can safely carry out the mounting, electrical installation, and maintenance work for the Flexi Loop safe series connection	8015834
Operating instructions	Flexi Loop in the Flexi Soft Designer configuration software	Description of how to configure and set the parameters for the Flexi Loop safe series connection using software	To provide technical personnel working for the machine manufacturer/operator with instructions so that they can safely configure and commission the Flexi Loop safe series connection	8014521
Operating instructions	Flexi Loop in Safety Designer Configuration software	Description of how to configure and set the parameters for the Flexi Loop safe series connection using software	To provide technical personnel working for the machine manufacturer/operator with instructions so that they can safely configure and commission the Flexi Loop safe series connection	8018174
Mounting instructions	Flexi Soft FX3-EBX3 and FX3-EBX4 Encoder/Motor Feedback Connection Boxes	Description of FX3-EBX3 and FX3-EBX4 encoder/motor feedback connection boxes	To provide technical personnel working for the machine manufacturer/operator with instructions so that they can safely carry out the mounting, electrical installation, commissioning, and maintenance work for FX3-EBX3 and FX3-EBX4 encoder/motor feedback connection boxes	8015600
Mounting instructions	Flexi Soft FX3-EBX1 Optimized Dual Encoder/Motor Feedback Connection Box	Description of the FX3-EBX1 optimized dual encoder/motor feedback connection box	To provide technical personnel working for the machine manufacturer/operator with instructions so that they can safely carry out the mounting, electrical installation, commissioning, and maintenance work for the FX3-EBX1 optimized dual encoder/motor feedback connection box	8019030

1.2 Scope

Product

These operating instructions apply to all Flexi-Soft gateways apart from the FX3-GEPR EFI-pro gateway, which can only be configured using the Safety Designer configuration software.

Document identification

Document part number:

- This document: 8014526
- Available language versions of this document: 8012483

You can find the current version of all documents at www.sick.com.

1.3 Target groups and structure of these operating instructions

These operating instructions cover how to configure the Flexi Soft gateways using the Flexi Soft Designer configuration software. They are intended for the following target groups: Project developers (planners, developers, designers), installers, electricians, programmers, operators and maintenance personnel.

1.4 Further information

www.sick.com

The following information is available via the Internet:

- Other language versions
- Data sheets and application examples
- CAD data for drawings and dimensional drawings
- Certificates (such as the EU declaration of conformity)
- Guide for Safe Machinery (six steps to a safe machine)

The following files are also available for download from this site:

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

1.5 Symbols and document conventions

The following symbols and conventions are used in this document:

Warnings and other notes

DANGER
Indicates a situation presenting imminent danger, which will lead to death or serious injuries if not prevented.

WARNING
Indicates a situation presenting possible danger, which may lead to death or serious injuries if not prevented.

CAUTION
Indicates a situation presenting possible danger, which may lead to moderate or minor injuries if not prevented.

NOTICE
Indicates a situation presenting possible danger, which may lead to property damage if not prevented.

NOTE
Highlights useful tips and recommendations as well as information for efficient and trouble-free operation.

Instructions to action

► The arrow denotes instructions to action.

1. The sequence of instructions for action is numbered.
2. Follow the order in which the numbered instructions are given.

✓ The check mark denotes the result of an instruction.

LED symbols

These symbols indicate the status of an LED:

○ The LED is off.
☼ The LED is flashing.
● The LED is illuminated continuously.

2 Safety information

2.1 General safety notes

Integrating the product

DANGER
The product can not offer the expected protection if it is integrated incorrectly.

- ► Plan the integration of the product in accordance with the machine requirements (project planning).
- ► Implement the integration of the product in accordance with the project planning.

Mounting and electrical installation

DANGER
Death or severe injury due to electrical voltage and/or an unexpected startup of the machine

- ► Make sure that the machine is (and remains) disconnected from the voltage supply during mounting and electrical installation.
- ► Make sure that the dangerous state of the machine is and remains switched off.

WARNING
Improper mounting or use

The target safety-related level may not be achieved in the event of non-compliance.

- ► When mounting, installing, and using the Flexi Soft safety controller, remember to observe all applicable standards and directives.
- ► Observe the relevant national and international legal provisions for the installation and use of the Flexi Soft safety controller, its commissioning, and technical inspections repeated at regular intervals.
- ► The manufacturer and operator of the machine on which the Flexi Soft safety controller is used are responsible for liaising with the relevant authorities about all applicable safety regulations/rules and for ensuring compliance with these.
- ► The notes, in particular the test notes, in these operating instructions (e.g. regarding use, mounting, installation, or integration into the machine controller) must always be observed.
- ► The thorough checks must be carried out by qualified safety personnel or specially qualified and authorized personnel, and must be recorded and documented by a third party to ensure that the tests can be reconstructed and retraced at any time.

Configuration

WARNING
Ineffectiveness of the protective device due to incorrect configuration

The dangerous state may not be stopped or not be stopped in a timely manner in the event of non-compliance.

The target safety-related level may not be achieved in the event of non-compliance.

- ► Check whether the configured safety application monitors the machine or plant as intended and if the safety of the configured application is maintained at all times. This must be ensured in every operating mode and secondary application. Document the results of this thorough check.
- ► Check the safety function again after any change to the configuration.
- ► Observe the testing information in the operating instructions for the connected protective devices.

Repairs and modifications

DANGER
Improper work on the product

A modified product may not offer the expected protection if it is integrated incorrectly.

- ► Apart from the procedures described in this document, do not repair, open, manipulate or otherwise modify the product.

2.2 Intended use

The Flexi Soft gateways can only be operated in conjunction with a Flexi Soft system.

The main module used must have a firmware version of at least V1.11.0; the Flexi Soft Designer configuration software must be at least V1.3.0.

The product is only suitable for use in industrial environments.

The product must only be used within the limits of the prescribed and specified technical specifications and operating conditions at all times.

Incorrect use, improper modification or manipulation of the product will invalidate any warranty from SICK; in addition, any responsibility and liability of SICK for damage and secondary damage caused by this is excluded.

2.3 Inappropriate use

The Flexi Soft gateways (FX0-Gxxx) do not support any of the security mechanisms that are required for communication within a safety network. The Flexi Soft gateways (FX0-Gxxx) are therefore not suitable for operation on a safety fieldbus. These Flexi Soft gateways only generate non-safety-related fieldbus data (status bytes) for control and diagnostic purposes.

The target safety-related level may not be achieved in the event of non-compliance.

- ► Never operate Flexi Soft gateways (FX0-Gxxx) on a safety fieldbus.

With the Flexi Soft gateways (FX0-Gxxx), it is possible to integrate non-safety-related data into the logic editor in such a way as to impair the safety function of the Flexi Soft system.

The dangerous state may not be stopped or not be stopped in a timely manner in the event of non-compliance.

- ► Do not use the Flexi Soft gateways (FX0-Gxxx) for safety-related applications.
- ► Before integrating a gateway into a Flexi Soft system, have this source of danger thoroughly checked by a safety specialist.

Exception: The FX3-GEPR EFI-pro gateway

The FX3-GEPR EFI-pro gateway also allows safety-related data to be exchanged.

2.4 Security information

Security advisories exist for these devices.

Observe the security advisories under: www.sick.com/psirt

2.5 Requirements for the qualification of personnel

The product must be configured, installed, connected, commissioned, and serviced by qualified safety personnel only.

Project planning

You need safety expertise to implement safety functions and select suitable products for that purpose. You need expert knowledge of the applicable standards and regulations.

Mounting, electrical installation and commissioning

You need suitable expertise and experience. You must be able to assess if the machine is operating safely.

Configuration

You need suitable expertise and experience. You must be able to assess if the machine is operating safely.

Operation and maintenance

You need suitable expertise and experience. You must be instructed in machine operation by the machine operator. For maintenance, you must be able to assess if the machine is operating safely.

3 Product description

The Flexi Soft gateways enable the Flexi Soft system to send non-safety-related data to external fieldbus systems for control and diagnostic purposes, and also to receive data from them.

NOTE
In these operating instructions, data exchange between the Flexi Soft system and the respective network is always viewed from the perspective of the network master (PLC). Consequently, data sent to the network by the Flexi Soft system is referred to as “input data” and data received from the network is referred to as “output data”.

An individual Flexi Soft gateway can only be operated on one Flexi Soft system. It does not have its own voltage supply. Two Flexi Soft gateways can be operated on one system at the same time.

The safety-related logic of the Flexi Soft system functions independently of the gateway. However, this is not the case if the Flexi Soft system has been configured in such a way that non-safety-related information from the fieldbus is integrated into the logic editor. In this case, availability problems may occur if the gateway is switched off.

The Flexi Soft gateways are configured using the Flexi Soft Designer configuration software. The computer with the configuration software can be connected with the Flexi Soft system either via the RS-232 interface of a main module, via USB or over Ethernet TCP/IP and an Ethernet gateway.

Detailed information on configuration of the Flexi-Soft system is contained in the operating instructions "Flexi Soft in the Flexi Soft Designer configuration software" (SICK part number 8012998).

3.1 Device variants

Important information

NOTE
If two computers establish TCP/IP connections to the same Flexi Soft main module of a Flexi Soft Ethernet gateway in parallel (e.g., via port 9000), the Flexi Soft main module will only communicate via the most recently established connection. As a result, the second computer will establish a further connection without closing the ones already established. There comes a point when too many connections to the computers are open via the gateway and the only messages being exchanged on those computers are messages for maintaining these connections (known as keep-alive messages). This causes the Flexi Soft system to switch to the “Serious error” state.

NOTE
The part numbers and type codes of the FX0-GENT V2 (1099830) and FX0-GPNT V2 (1099832) displayed in Flexi Soft Designer deviate from the type label of the devices up to version 1.9.0 SP2 of Flexi Soft Designer.

The part numbers and type codes of the FX0-GMOD V2 (1130282) displayed in Flexi Soft Designer deviate from the type label of the devices up to version 1.9.6 SP1 of Flexi Soft Designer.

Device variants

Table 2: Device variants and their main features

Gateway	Network type	Ethernet TCP/IP socket interface	TCP/IP configuration interface
FX0-GENT	EtherNet/IP™ with explicit messaging	Client/server	TCP port 9000 UDP port 30718
FX0-GMOD	Modbus TCP with master and slave operation	Client/server	TCP port 9000 UDP port 30718
FX0-GPNT	PROFINET IO slave, conformance class A	Client/server	TCP port 9000 UDP port 30718
FX0-GETC	EtherCAT slave	–	TCP port 9000 and UDP port 30718 via EoE
FX0-GPRO	PROFIBUS DP slave	–	–
FX0-GCAN	CANopen slave	–	–
FX0-GDEV	DeviceNet slave	–	–

2)) The TCP/IP configuration interface for the FX0-GETC will only be available if EoE (Ethernet over EtherCAT) has been configured in advance.

Complementary information

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

3.2 Firmware versions

The FX0-GENT, FX0-GMOD, and FX0-GPNT Ethernet gateways and the FX0-GDEV DeviceNet gateway are available with a variety of firmware versions. In order to add a gateway to a Flexi Soft system in the configuration software, you have to select the appropriate step of the respective gateway.

Table 3: Firmware versions of the Ethernet gateways

Firmware version	Step
V1.xx.x	1.xx
V2.xx.x	2.xx
≥ V3.00.0	3.xx

NOTE

- You will find the firmware version on the device type label.
- When you use the configuration software to read in a Flexi Soft system, the firmware version of the devices is detected automatically.

4 Configuration

4.1 Ethernet gateways

This chapter describes how to configure the following gateways:

- The FX0-GENT EtherNet/IP™ gateway
- FX0-GMOD Modbus TCP gateway
- FX0-GPNT PROFINET IO gateway
- FX0-GETC EtherCAT gateway

4.1.1 The FX0-GENT EtherNet/IP™ gateway

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

4.1.1.1 Basic configuration – Assigning device name and IP address

► Start Flexi Soft Designer and load the hardware configuration, including the EtherNet/IP™ gateway.
► Click on the **Interfaces** button above the main window and select the FX0-GENT or double-click on the FX0-GENT in the **Hardware configuration** to open the dialog box for gateway configuration.
► Click on the **Gateway configuration** button. The following dialog box is displayed:

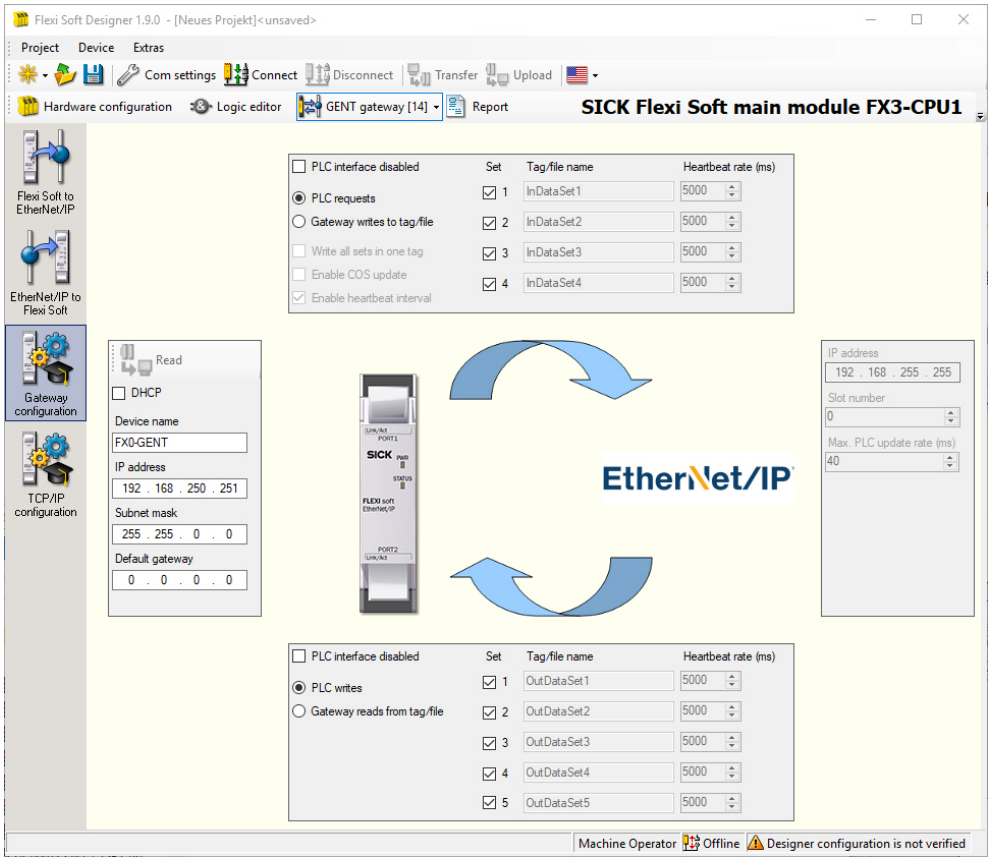

Figure 1: Configuration window for the EtherNet/IP™ gateway

► If you wish, change the **Device name** of the gateway.
► Enter an **IP address** for the gateway and, if necessary, a **Subnet mask** and an IP address for a **Default gateway**.
► Click on **Connect** to switch to online mode.
► Click on **Transfer** to transfer the configuration to the Flexi Soft system.

4.1.1.2 EtherNet/IP™ Class 1 communication – implicit messaging

NOTE

- Only Flexi Soft EtherNet/IP™ gateways with a firmware version ≥ V2.00.0 support both EtherNet/IP™ implicit messaging (Class 1) as well as explicit messaging (Class 3).
- Flexi Soft EtherNet/IP™ gateways with a firmware version < V2.00.0 only support explicit messaging (Class 3).
- You will find the firmware version on the device type label.

General overview

Implicit messaging is a method of communication between EtherNet/IP™-enabled controllers (PLC) and corresponding devices.

- Implicit messaging uses Ethernet UDP telegrams.
- Implicit messaging is cyclic. Input and output data is exchanged at regular time intervals between controls and devices.
- Delivery of the telegrams is not guaranteed.
- Multicast addressing is possible.

Functional principle of Class 1

To establish Class 1 communication, a connection must be created between the PLC and the EtherNet/IP™ gateway. For this purpose, the PLC must send a Forward Open message to the gateway. The gateway then checks the received parameters and, depending on the success or failure, responds with a status message (Forward Open response) which, if successful, contains a set of connection parameters.

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

- Connection type for the input data (gateway to the PLC: either point-to-point or multicast)
- Instance number of the Input Assembly object
- Length of the input data
- Connection type for the output data (PLC to the gateway: only point-to-point connections will be accepted.)
- Instance number of the Output Assembly object (not used if only input data is sent)
- Length of the output data (not used if only input data is sent)
- Requested Packet Interval

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

- Status of the connection attempt
  - SUCCESS: The connection is permitted if the received parameters can be accepted and the gateway has sufficient bandwidth and memory. The STATUS LED lights up ● Green.
  - FAILURE: The connection attempt will be rejected if either the received parameters are incorrect or the gateway does not have sufficient bandwidth or memory. The STATUS LED flashes ☼ Red/green.
- The IP address and the UDP socket number at which the PLC must expect incoming telegrams:
  - This is the IP address of the PLC in the case of a point-to-point connection for the input data.
  - In the case of a multicast connection for the input data, the gateway sends the multicast address at which the PLC must expect incoming telegrams.

- The packet interval allowed by the gateway. This can be equal to or greater than the interval requested by the PLC.

Once a connection has been successfully established, 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

FX0-GENT with firmware ≥ V3.00.0:

The packet update interval for Class 1 connections that is returned in the Forward Open response to the EtherNet/IP™ PLC depends on the received value for the **Requested packet interval** in the Forward Open message of the EtherNet/IP™ PLC. It can be set in increments of 1 ms. The minimum value is 4 ms.

FX0-GENT with firmware < V3.00.0:

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

- The received value for the **Requested packet interval** in the Forward Open message of the EtherNet/IP™ PLC
- The 10 ms system clock that the EtherNet/IP™ gateway operates on
- The **Max. PLC update rate** which was set in the configuration software for the gateway

If the requested packet interval is lower than the **Max. PLC update rate**, then the packet update interval is set to the **Max. PLC update rate**. Otherwise, it is set to the requested packet interval value.

If the Packet Update Interval is not a multiple of 10 ms (10, 20, 30, 40, etc.), it will be rounded up to the next multiple of 10 ms.

Table 4: Examples for the Packet Update Interval

Requested Packet Interval	Max. PLC update rate	Actual Packet Update Interval	Description
5 ms	10 ms	10 ms	Set to max. PLC update rate
10 ms	10 ms	10 ms	Requested Packet Interval accepted
15 ms	20 ms	20 ms	Set to max. PLC update rate
15 ms	10 ms	20 ms	Requested Packet Interval rounded up to 20 ms
20 ms	25 ms	30 ms	Max. PLC update rate rounded to 30 ms
40 ms	30 ms	40 ms	Requested Packet Interval accepted
32 ms	30 ms	40 ms	Requested Packet Interval rounded up to 40 ms
48 ms	40 ms	50 ms	Requested Packet Interval rounded up to 50 ms
50 ms	40 ms	50 ms	Requested Packet Interval accepted

Bandwidth limits

The maximum number of Class 1 telegrams per second is limited by the main module. If 50% of the main module bandwidth is available, this corresponds to approximately 200 telegrams per second or a Class 1 connection with an update rate of 10 ms (the system clock of the FX0-GENT is 10 ms).

Table 5: Recommended bandwidths for Class 1 telegrams

Update rate of the PLC [ms]	Cyclic I/O connections in two directions	Cyclic Input Only multicast connections
10	1	2
20	2	4
40	Up to 4	Up to 8

NOTE

The gateway does not force these recommended bandwidths. However, if the bandwidth used for Class 1 communication exceeds 200 telegrams per second, this will slow down the RS-232 and the Ethernet TCP/IP interfaces.

Point-to-point connections and multicast connections

For EtherNet/IP™ PLC to gateway:

- Only point-to-point connections are accepted.
- Multicast connections are not accepted.

For gateway to EtherNet/IP™ PLC:

- Both point-to-point connections and multicast connections are accepted.
- Multicast connections can be established either with an exclusive receiver or several receivers.

Maximum number of Class 1 connections

FX0-GENT with firmware ≥ V3.00.0:

The FX0-GENT supports a total number of eight connections.

FX0-GENT with firmware < V3.00.0:

- For I/O connections involving both input and output data sets:
  - Each output data set can only be controlled by one I/O connection.
  - Attempts by an I/O connection to control an output data set are rejected if the output data set is already being controlled by another I/O connection.
  - If each I/O connection controls just one output data set, up to five 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 bandwidth, up to 32 connections can be established simultaneously, if all are requesting the same input data set.
  - The maximum number of connections for requesting different data depends on the data transmission rates requested and the gateway bandwidth available. If the connections need more bandwidth than is available, the gateway will slow down and become unable to output the input data at the requested data transmission rates.

Class 1 access to input data sets

- All four input data sets are contained in an array that can be read out by all Class 1-capable controls.
- The start of the input data received by the PLC is determined 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 the PLC to receive only part of an input data set or also several input data sets. For example, the PLC could read only the first 20 bytes of input data set 1 or all input data sets.
- All input data that is sent to the PLC must follow in direct succession. In other words, input data sets 1 and 2 or input data sets 1, 2 and 3 can be sent jointly. However, input data sets 1 and 3 cannot be sent together as they do not follow each other directly.

Table 6: Class 1 read access points to input data sets

Assembly instance	Byte index	Length [bytes]	Input data set	Description	Valid lengths for read access [bytes]
1	0 ... 49	50	1	Starts with input data set 1 Can read input data sets 1 to 4	1 ... 202
2	50 ... 81	32	2	Starts with input data set 2 Can read input data sets 2 to 4	1 ... 152
3	82 ... 141	60	3	Starts with input data set 3 Can read input data sets 3 and 4	1 ... 120
4	142 ... 201	60	4	Can read input data set 4	1 ... 60

Class 1 access to output data sets

- All five output data sets are contained in an array that can be written to by all Class 1-capable controls.
- The start of the output data is determined by the assembly instance number. Each output instance number corresponds to the start of an output data set.
- The length determines how many output data items are sent by the PLC. This allows the PLC to write to only one output data set or also several output data sets. For example writing is possible only to output data set 1 or all five output data sets.
- If it is not possible to write to only parts of an output data set, the length for output data set must be a multiple of 10 bytes. The length must be 10 bytes in order to write one output data set, 20 for two output data sets, etc.
- All output data sets to which the PLC is to write simultaneously must follow in direct succession. This means that it is possible to write simultaneously to output data sets 1 and 2 or output data sets 1, 2 and 3, for example. However, it is not possible to write to output data sets 1 and 3 simultaneously as they do not follow each other directly.

Table 7: Class 1 write access points to output data sets

Assembly instance	Byte index	Length [bytes]	Output data set	Description	Valid lengths for write access [bytes]
5	0 ... 9	10	1	Starts with output data set 1 Can write output data sets 1 to 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	Starts with output data set 2 Can write output data sets 2 to 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	Starts with output data set 3 Can write output data sets 3 to 5	10 = Output data set 3 20 = Output data sets 3 ... 4 30 = Output data sets 3 ... 5
8	30 ... 39	10	4	Starts with output data set 4 Can write output data sets 4 and 5	10 = Output data set 4 20 = Output data sets 4 ... 5
9	40 ... 49	10	5	Start at output data set 5 Can write to output data set 5	10 = Output data set 5

Description of the assembly object

All Class 1 data must be transferred using the assembly object. The assembly object is used as an interface to directly link manufacturer-specific objects to a standard interface, which the EtherNet/IP™-enabled 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. Each instance of the assembly object corresponds to one or more full data set transfer object attributes.

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

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

Table 8: Class attributes of the assembly object

Attribute ID	Name	Data type	Data values	Access type
1	Revision	UINT	1	Read
2	Max. instance	UINT	9	Read
3	Number of instances	UINT	9	Read

Table 9: Description of the assembly object instances

Assembly instance	Description	Data type	Data values	Access type	Corresponding attributes of the Full Data Set Transfer object
Flexi Soft to the network					
1	Request data of input data sets 1 to 4	BYTE[202] Valid lengths for read access: 1 ... 202	0 ... 255	Read	1, 2, 3, 4
2	Request data of input data sets 2 to 4	BYTE[152] Valid lengths for read access: 1 ... 152	0 ... 255	Read	2, 3, 4
3	Request data of input data sets 3 and 4	BYTE[120] Valid lengths for read access: 1 ... 120	0 ... 255	Read	3, 4
4	Request data of input data set 4	BYTE[60] Valid lengths for read access: 1 ... 60	0 ... 255	Read	4
Network to the Flexi Soft					
5	Write data to output data sets 1 to 5	BYTE[50] Valid lengths for write access: 10 = Data set 1 20 = Data sets 1 ... 2 30 = Data sets 1 ... 3 40 = Data sets 1 ... 4 50 = Data sets 1 ... 5	0 ... 255	Read/ Write	5, 6, 7, 8, 9
6	Write data to output data sets 2 to 5	BYTE[40] Valid lengths for write access: 10 = Data set 2 20 = Data sets 2 ... 3 30 = Data sets 2 ... 4 40 = Data sets 2 ... 5	0 ... 255	Read/ Write	6, 7, 8, 9
7	Write data to output data sets 3 to 5	BYTE[30] Valid lengths for write access: 10 = Data set 3 20 = Data sets 3 ... 4 30 = Data sets 3 ... 5	0 ... 255	Read/ Write	7, 8, 9

Assembly instance	Description	Data type	Data values	Access type	Corresponding attributes of the Full Data Set Transfer object
8	Write data to output data sets 4 and 5	BYTE[20] Valid lengths for write access: 10 = Data set 4 20 = Data sets 4 ... 5	0 ... 255	Read/ Write	8, 9
9	Write data to output data set 5	BYTE[10] Valid lengths for write access: 10 = Data set 5	0 ... 255	Read/ Write	9

Table 10: Instance attributes of the assembly object

Attribute ID	Name	Data type	Data values	Access type
3	Data	BYTE array	0 ... 255	Read/Write
4	Data length	UINT	Maximum number of bytes in attribute 3	Read

Attribute 3 – Request/write data: either the input data to be read or the output data to be written, depending on the instance number

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

Common services

Table 11: Common services of the assembly object

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_Attributes_All

Example configuration of implicit messaging with Rockwell RSLogix 5000

A description of configuration of a Class 1 connection with Rockwell RSLogix 5000 can be found in the brochure "Flexi Soft EtherNet IP: Implicit Messaging with Rockwell RSLogix 5000" (SICK part number 8015358). This brochure is available for download as a PDF file at www.sick.com.

Example of how to configure implicit messaging with an OMRON PLC

The brochure titled “Flexi Soft EtherNet IP: Implicit Messaging with an OMRON PLC” (SICK part number 8015333) contains a description of how to configure a Class 1 connection with an OMRON PLC. This brochure is available to download as a PDF from www.sick.com.

4.1.1.3 EtherNet/IP Class 3 communication – Explicit messaging

General overview

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

- Explicit messaging uses Ethernet TCP/IP telegrams.
- Explicit messaging is not cyclic. The PLCs and devices must send individual telegrams to each other.
- Delivery of the telegrams is guaranteed.
- Multicast addressing is not possible.

Important information

NOTE
The EDS file V3.01 for the FX0-GENT is not compatible with an OMRON control.

Transmission modes

The configuration steps in this section determine the way in which the data is transmitted to the higher-level PLC. In general, there are two different transmission types for both directions, i.e., for Flexi Soft to the network and network to the Flexi Soft:

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

Both types can be merged. For example, it is possible to configure the gateway as a master for the Flexi Soft to network transmission direction (**Gateway writes to tag/file** option selected), while at the same time it operates as a slave for the network to Flexi Soft direction (**PLC writes** option selected).

Number of possible connections

FX0-GENT with firmware ≥ V3.00.0:

The FX0-GENT supports a total number of eight connections.

FX0-GENT with firmware < V3.00.0:

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

Table 12: Number of possible connections

Transmission type	Maximum number of connections
Rx (to PLC): Gateway writes to tag/file (master) Tx (from PLC): Gateway reads from tag/file (master)	Rx and Tx: 1
Rx (to PLC): Gateway writes to tag/file (master) Tx (from PLC): PLC writes (slave)	Rx: 1 Tx: 127
Rx (to PLC): PLC requests (slave) Tx (from PLC): Gateway reads from tag/file (master)	Rx: 127 Tx: 1
Rx (to PLC): PLC requests (slave) Tx (from PLC): PLC writes (slave)	Rx and Tx: 128

Configuration

The following table describes the configuration process depending on the transmission mode.

Table 13: Configuration guideline – Gateway as master

Gateway is master (Gateway writes to Tag/File and/or Gateway reads from Tag/File)	
Necessary settings in the gateway configuration	**Required settings in the PLC program and/or in the EtherNet/IP™ configuration tool**
Select which data is to be written to or read from the PLC.	–
Define the location in the PLC memory to which the selected data shall be written: enter tag names. Example: InDataSet1 Define the location in the PLC memory from which the selected data shall be read: enter tag names. Example: OutDataSet1	Enter exactly the same tag names in the PLC program. Example: InDataSet1 INT[25] OutDataSet1 INT[5] The data type must be INT.
Select how often this data shall be transmitted.	–
Determine the location in the EtherNet/IP™ network from which and to which the data is to be read or written: Enter the IP address and slot number of the PLC controller.	–

Table 14: Configuration guideline – Gateway as slave

Gateway is a slave (PLC requests and/or PLC writes)	
Necessary settings in the gateway configuration	**Required settings in the PLC program and/or in the EtherNet/IP™ configuration tool**
–	Download and install the EDS file for the FX0-GENT from www.sick.com. The EDS file is also contained in the program directory of the SICK configuration software.
–	Connect the FX0-GENT to the EtherNet/IP™ network using a network configuration tool (e.g., RSNetworx).
–	Use the explicit message "Get_Attribute_..." or "Set_Attribute_..." in the program of the PLC to read data from the gateway or write data to the gateway.
–	Program the trigger for sending explicit messages.

Transmission mode 1: Gateway writes to/reads from tag/file – The FX0-GENT writes the data to/reads the data from the memory of the PLC.

In this transmission mode, the FX0-GENT as master writes the data of all activated data sets to the specified memory areas of the PLC. The only task of the PLC programmer is to define a tag name in the control that corresponds to the tag name in the gateway configuration.

Configuring the gateway as a master

- ► Start Flexi Soft Designer and load the hardware configuration, including the EtherNet/IP™ gateway.
- ► Click on the **Interfaces** button above the main window and select the FX0-GENT or double-click on the FX0-GENT in the **Hardware configuration** to open the dialog box for gateway configuration.
- ► Click on the **Gateway configuration** button. The following dialog box is displayed:

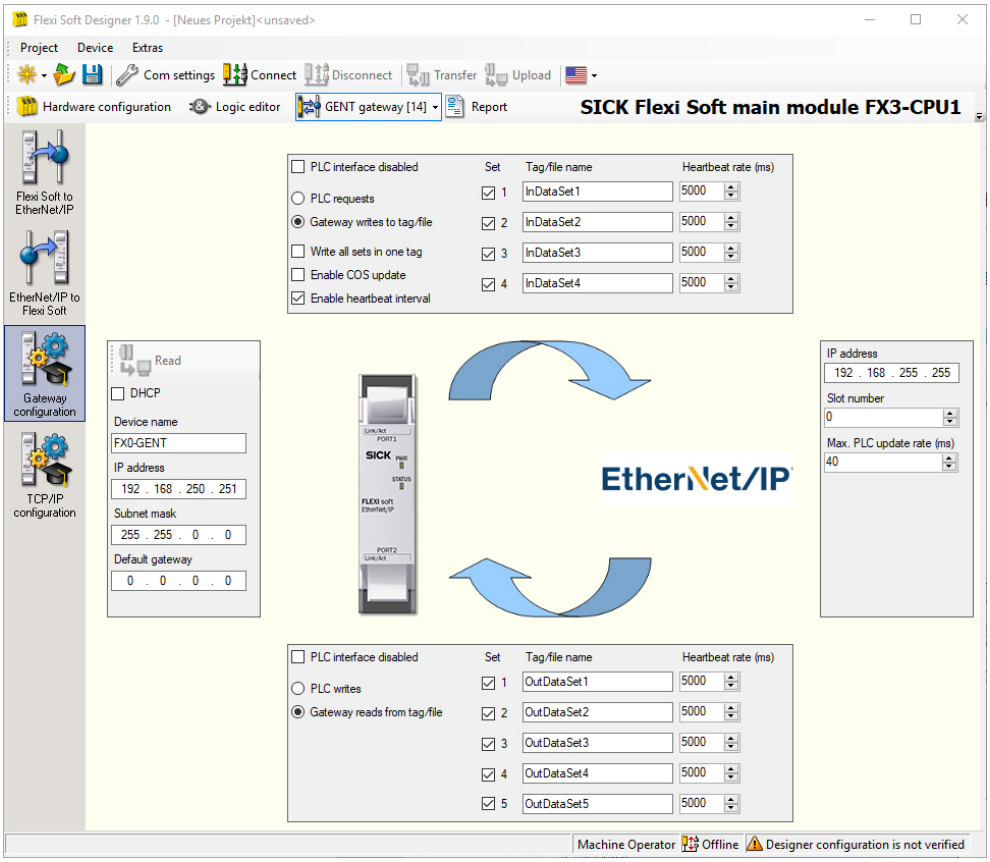

Figure 2: Configuring the EtherNet/IP™ gateway as a master

- ► Select the transmission type in the **Gateway configuration** dialog box: Select the **Gateway writes to tag/file** option button for the Flexi Soft to network direction. Select the **Gateway reads from tag/file** option button for the network to Flexi Soft direction.
- ► Check the boxes for the required data sets in the relevant configuration area to select the data which is to be written to the PLC or read from it.
- ► Specify the location in the PLC memory to which or from which the selected data is to be written or read. Enter tag names in the **Tag/file name** input fields (max. 20 characters).
- ► For the Flexi Soft to network transmission direction, select the **Write all sets in one tag** option if all data sets are to be written to the PLC memory in a single tag.
- ► For the Flexi Soft to network transmission direction, specify how often the data is to be transferred to the PLC:
  - Select the **Enable COS update** (update in the event of change of state) option if the FX0-GENT must update the data in the PLC immediately if something in the data sets changes.
  - Select the **Enable heartbeat rate** option to update the selected data sets cyclically with the set **Heartbeat rate** in milliseconds.
  - Both boxes can be checked at the same time.
- ► For the network to Flexi Soft transmission direction, specify how often the data is to be read from the PLC:
  - Enter a **Heartbeat rate** in milliseconds to update the selected data sets at the specified interval.
- ► Specify the location in the EtherNet/IP™ network to which or from which the selected data is to be written or read. Enter the **IP address** and **Slot number** of the PLC controller.

NOTE

The configuration is incorrect if the PLC IP address is zero and **Gateway writes to tag/file** is selected for the Flexi Soft to network direction and/or **Gateway reads from tag/file** for the network to Flexi Soft direction.

- The **Max. PLC update rate** determines the maximum rate for transferring data sets to or from the PLC. The setting is dependent on the processing speed of the PLC and can be between 10 and 65,535 ms. The default setting of 40 ms is suitable for most PLCs.

NOTE

- If the value entered for the **Max. PLC update rate** is higher than the **Heartbeat rate** set for writing to or reading from the PLC, then the heartbeat rate is automatically increased to this value (i.e., slowed down).
- All data sets are transferred to the PLC in integer format (16-bit word), whereby the first byte is positioned as the most significant or leftmost byte of the integer.

- Click on **Connect** to switch to online mode.
- Click on **Transfer** to transfer the configuration to the Flexi Soft system.
- Start the configuration software for the PLC.
- Define the tag names in the PLC in the way that these were configured previously in the Flexi Soft EtherNet/IP™ gateway. The figure below shows an example of defining tag names in a PLC program, which was written with RSLogix:

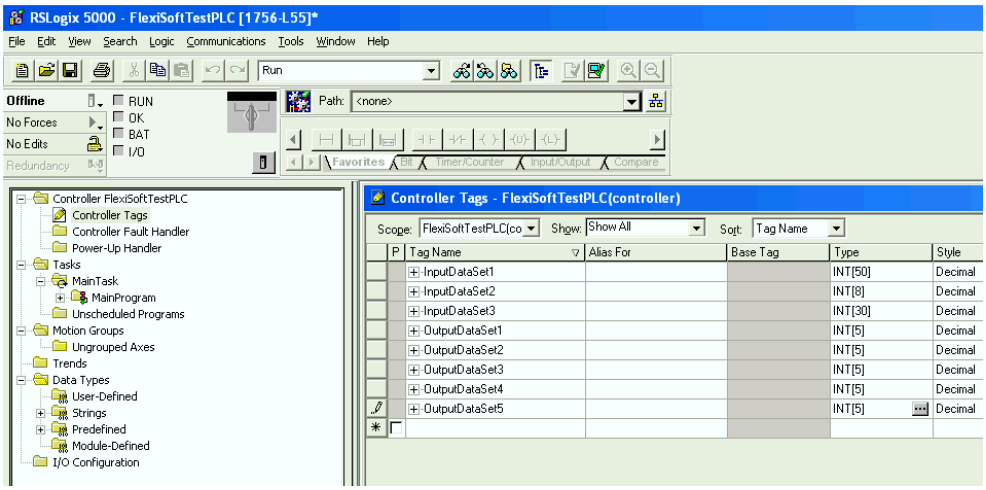

Figure 3: Example of tag names in a PLC program

NOTE

- Tag names of an Allen Bradley SLC/PLC-5 PLC must start with a "$" (e.g. $N10:0).
- Tag names of an Allen Bradley MicroLogix PLC must start with a "#" (e.g. #N10:0).

Transmission mode 2: Polling mode – The PLC requests data or writes data to the FX0-GENT.

With this transmission type, the FX0-GENT works as a slave. It sends data to the PLC on request and the PLC writes the data to the gateway.

Configuring the gateway as a slave

- ► Start Flexi Soft Designer and load the hardware configuration, including the EtherNet/IP™ gateway.
- ► Click on the **Interfaces** button above the main window and select the FX0-GENT or double-click on the FX0-GENT in the **Hardware configuration** to open the dialog box for gateway configuration.
- ► Click on the **Gateway configuration** button. The following dialog box is displayed:

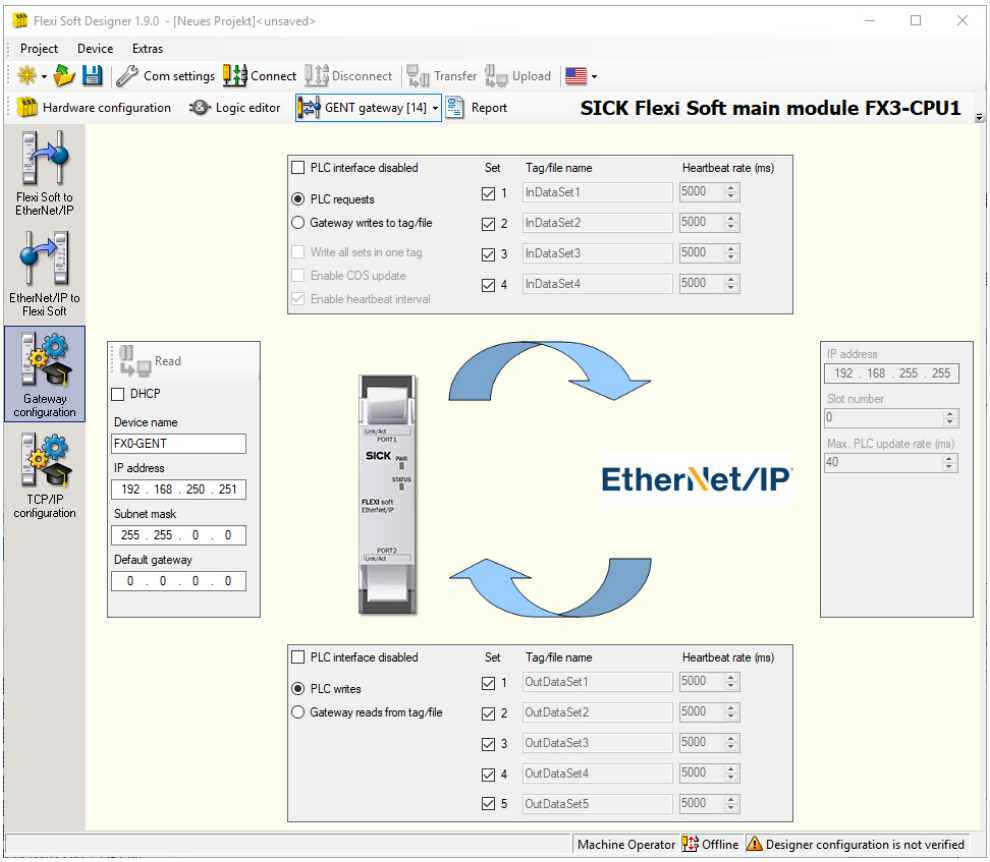

Figure 4: Configuring the EtherNet/IP™ gateway as a slave

- ► Select the transmission type in the **Gateway configuration** dialog box: Select the **PLC requests** option button for the Flexi Soft to network direction. Select the **PLC writes** option button for the network to Flexi Soft direction.
- ► Check the boxes for the required data sets in the relevant configuration area to select the data which is to be requested or written by the PLC.
- ► Click on **Connect** to switch to online mode.
- ► Click on **Transfer** to transfer the configuration to the Flexi Soft system.
- ► Program explicit messaging in the PLC.

Polling data sets via explicit messaging

The FX0-GENT supports two manufacturer-specific objects that can be polled with explicit messaging:

- The **Full Data Set Transfer** allows each individual data set to be requested. There is one instance, whereby each attribute represents a data set.
- The **Individual Input Data Set Transfer** object allows the individual data set parameters to be requested. There is one instance per data set and each attribute represents one byte of the data set.

Object definition for Full Data Set Transfer (72h – one instance)

The manufacturer-specific Full Data Set Transfer object defines the attributes with which the PLC can perform the following actions:

- request complete input data sets from the FX0-GENT
- write complete output data sets to the FX0-GENT

Class attributes (instance 0)

Table 15: Class attributes (instance 0) for the object Full Data Set Transfer (72h)

Attribute ID	Name	Data type	Data value(s)	Access type
1	Revision	UINT	1	Read
2	Max. instance	UINT	1	Read
3	Number of instances	UINT	1	Read

Instance attributes (instance 1)

The instance attributes permit access to the input and output data sets. **Get Attribute Single** requests for a specific input data set return its input data set information. **Get Attributes All** requests return all activated input data sets.

All data set information is returned in integer format (16-bit word). In the case of byte-oriented 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.

Table 16: Instance attributes (instance 1) for the object Full Data Set Transfer (72h)

Attribute ID	Name	Data type	Data value(s)	Access type
Flexi Soft to the network				
1	Request of input data set 1-specific data	UNIT array	0 ... 255	Read
2	Request input data set 2-specific data	UNIT array	0 ... 255	Read
3	Request input data set 3-specific data	UNIT array	0 ... 255	Read
4	Request input data set 4-specific data	UNIT array	0 ... 255	Read
Network to Flexi Soft				
5	Write output data set 1-specific data	UNIT array	0 ... 255	Write
6	Write output data set 2-specific data	UNIT array	0 ... 255	Write
7	Write output data set 3-specific data	UNIT array	0 ... 255	Write
8	Write output data set 4-specific data	UNIT array	0 ... 255	Write
9	Write output data set 5-specific data	UNIT array	0 ... 255	Write

Common services

Table 17: Common services for the object Full Data Set Transfer (72h)

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_Attributes_All

Object Individual Input Data Set Transfer (73h – one instance per data set)

The manufacturer-specific object **Individual Input Data Set Transfer** defines the attributes with which the PLC can request both complete input data sets as well as individual parameters within a data set.

Class attributes

Table 18: Class attributes for the object Individual Input Data Set Transfer (73h)

Attribute ID	Name	Data type	Data value(s)	Access type
1	Revision	UINT	1	Read
2	Max. instance	UINT	4	Read
3	Number of instances	UINT	4	Read

Instance attributes

Table 19: Instance attributes for the object Individual Input Data Set Transfer (73h)

Attribute ID	Name	Data type	Data value(s)	Access type
1 to n (depending on the definition of the data set)	Request input data set-specific data	SINT	0 ... 255	Read

Common services

Table 20: Common services for the object Individual Data Set Transfer (73h)

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

Definitions of instance attributes

Attributes 1 to n – Request input data-specific parameters
Attributes 1 to n return the input data set-specific data arrays. **Get Attribute Single** requests for a specific input data set return only the parameter information for the requested data set. **Get Attributes All** requests return the entire data set.

The attributes numbered consecutively from 1 to n refer to the individual attributes of the respective 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 show the attribute definitions for each input data set.

Get All Data Set Attributes request

All data set information is returned in integer format (16-bit word). In the case of byte-oriented 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 is returned as follows:

- IntegerArray[0]: BBAAh – AA = BYTE1; BB = BYTE2
- IntegerArray[1]: DDCCh – CC = BYTE3; DD = BYTE4
- ...
- IntegerArray[6]: NNMMh – MM = BYTE13; NN = BYTE14

NOTE

The usual tools of Rockwell/Allen Bradley change this data format back to BBAA hex format for visualization purposes. The plausibility of the transmitted data must therefore be checked before putting a Flexi Soft system into operation.

Instance 1 – Attribute definitions for input data set 1

Table 21: Attribute definitions for instance 1 of object Individual Input Data Set Transfer (73h)

Attribute ID	Data set parameters	Size
1	Byte 0	SINT
2	Byte 1	SINT
...	...	...
50	Byte 49	SINT

Instance 2 – Attribute definitions for input data set 2

Table 22: Attribute definitions for instance 2 of object Individual Input Data Set Transfer (73h)

Attribute ID	Data set parameters	Size
1 ... 4	Overall checksum	UDINT
5 ... 8	Flexi Soft checksum	UDINT
9 ... 12	FX3-CPU0 and FX3-CPU1: reserved FX3-CPU2 and FX3-CPU3: ACR checksum	UDINT
13 ... 16	Reserved	UDINT
17 ... 20	Reserved	UDINT
21 ... 24	Reserved	UDINT
25 ... 28	Reserved	UDINT
29 ... 31	Reserved	UDINT

Instance 3 – Attribute definitions for input data set 3

Table 23: Attribute definitions for instance 3 of object Individual Input Data Set Transfer (73h)

Attribute ID	Data set parameters	Size
1	Status Module 0	UINT[2]
2	Status Module 1	UINT[2]
...	...	...
15	Status Module 14	UINT[2]

Instance 4 – Attribute definitions for input data set 4

Table 24: Attribute definitions for instance 4 of object Individual Input Data Set Transfer (73h)

Attribute ID	Data set parameters	Size
1	Reserved	UINT[2]
2	Reserved	UINT[2]
...	...	...
15	Reserved	UINT[2]

SLC, PLC-5 and MicroLogix PLC interfaces

FX0-GENT with firmware version < 3.00.0 supports SLC, PLC-5, and MicroLogix PLC interfaces:

- The same write-to-PLC functionality as with the **Write to file** reception type for ControlLogix PLC
- PCCC-based messages transferred with 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 logical binary address format)
- Normal SLC and PLC-5 file naming conventions are used.

The major differences between the SLC, PLC-5, and MicroLogix PLC interfaces and the ControlLogix interfaces are as follows:

- Polling is performed through the SLC- and PLC-5-specific messages instead of accessing one of the Data Transfer objects.
- Data is written to files on the PLC and not to tags as on a ControlLogix PLC.

NOTE

ControlLogix PLCs do support SLC messages and PLC-5 messages, but for reasons relating to data volume and performance, it is not advisable to use these messages on ControlLogix PLCs.

FX0-GENT with firmware version ≥ 3.00.0 does not support SLC, PLC-5, and MicroLogix PLC interfaces.

Receive modes

- **Polling**
  This receive mode allows the PLC to regularly request data by means of polling. With this receive mode, the input data set information is returned in the response to the data request message. The PLC requests the data by accessing the corresponding address on the FX0-GENT either with a SLC typed or PLC-5 typed read message.
  The following limitations apply:
  - The storage location on the PLC that is to receive the input data set must be of the type Integer and must be large enough for the input data set table(s).
  - If no data has been received for the specified module on the FLEXBUS+, only zeros will be returned.
- Unsolicited – **Write to file**
  If the data received by the Flexi Soft gateway via the FLEXBUS+ interface is to be sent to the PLC, the data will be immediately written to a file on the PLC.
  The following limitations apply:
  - The file name for the receive data area must have the same name as the file defined on the PLC. For SLC and PLC-5 PLCs, all file names must start with "$" (e.g. $N10:0). For MicroLogix PLCs, all file names must start with "#" (e.g. #N10:0).
  - The file on the PLC must be of the type Integer and must be large enough for the input data set table(s).
  - The data is written so that the first byte is placed in the MS byte (most significant, leftmost) location of the integer.
    Example: aabb, ccdd, eeff, etc., where aa = byte 1, bb = byte 2, cc = byte 3, etc.

Transmit data transfer modes (from the PLC)

The FX0-GENT supports the following transmission types for receiving or requesting output data sets from the PLC:

- **PLC writes**
  This is the standard transmission mode where the PLC uses a message to write the output data sets to the FX0-GENT. With this transmission mode, output data sets can be updated via a PCCC message to the corresponding file or address location on the FX0-GENT.
- **Read from file** (polling from the PLC)
  In this transmission mode, the FX0-GENT monitors the configured PLC memory area for changes in the output data sets. The output data sets will be processed accordingly if a change is detected.
  The following limitations apply:
  - The storage location that is to receive the output data set must be of the type Integer (16-bit word format) and must be large enough for the entire output data set.
  - The data in the integer file must be formatted so that the first byte is placed in the MS byte (most significant, leftmost) location.
    Example: aabb, ccdd, eeff, etc., where aa = byte 1, bb = byte 2, cc = byte 3, etc.

SLC and PLC-5 messages

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

Table 25: Supported PCCC messages for SLC, PLC-5, 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 SLC and PLC-5 Typed Read messages can be used to request all input data sets.

Table 26: Addressing for SLC and PLC-5 messages

Address	Description	Access type	Data size [words]
N10:0	Request data of all activated input data sets	Read	16 ... 101 [1]
N11:0	Request data of input data set 1	Read	25
N12:0	Request data of input data set 2	Read	16
N13:0	Request data of input data set 3	Read	30
N14:0	Request data of input data set 4	Read	30
N20:0	Write all activated input data sets	Write	5 ... 25 [2]
N21:0	Write data of output data set 1	Write	5
N22:0	Write data of output data set 2	Write	5
N23:0	Write data of output data set 3	Write	5
N24:0	Write data of output data set 4	Write	5

Address	Description	Access type	Data size [words]
N25:0	Write data of output data set 5	Write	5

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

SLC/PLC-5 Message – Data reception

The format of the received input data set message is defined for each individual input data set (detailed description: see "Data transferred to the network (network input data sets)", page 128).

PCCC object (67h – 1 instance)

The **PCCC** object can receive PCCC messages and transfer them between devices in an EtherNet/IP™ network. This object is used to communicate with SLC-5/05 and PLC-5 PLCs over EtherNet/IP™.

Class attributes

Not supported.

Instance attributes

Not supported.

Instances

Supports instance 1.

Common services

Table 27: Common services of the PCCC object (67h)

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

Message structure for Execute_PCCC

Table 28: PCCC object (67h) request message

Name	Data type	Description
Length	USINT	Length of ID of the requesting device
Manufacturer	UINT	Manufacturer of the requesting device
Serial number	UDINT	ASA serial number of the requesting device
CMD	USINT	Command byte
STS	USINT	0
TNSW	UINT	Transport word
FNC	USINT	Function code
PCCC_params	USINT array	CMD-/FNC-specific parameters

Table 29: PCCC object (67h) response message

Name	Data type	Description
Length	USINT	Length of ID of the requesting device
Manufacturer	UINT	Manufacturer of the requesting device
Serial Number	UDINT	ASA serial number of the requesting device
CMD	USINT	Command byte
STS	USINT	Status byte
TNSW	UINT	Transport word, the same value as request

Name	Data type	Description
EXT_STS	USINT	Extended status (if error)
PCCC_params	USINT array	CMD-/FNC-specific result data

Table 30: PCCC commands supported by the PCCC object (67h)

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

Example configuration of explicit messaging

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

Required gateway settings

Select the following option buttons in the Gateway configuration of the Flexi Soft configuration software:

- PLC requests
- PLC writes

Required RSLogix settings

Make the following settings in RSLogix:

- PLC is active (explicit messaging activated)
- 128 possible connections
- Each data set must have the correct size.
- The main program sends a message with a SET or GET command (e.g. Get_Attribute_Single or Get_Attributes_All).

Step 1:

► Create two tags: one for the message and one to store the data from the GET command.

The **message** tag is intended for the MSG block which is used for explicit messaging. The MSG command in this example requests Data Set 1. The received Data Set 1 is then places in a self-defined tag with the name **WhatIWant**.

NOTE

The destination tag must be configured with the same size as the data set. In this case, this is 50 bytes or INT[25].

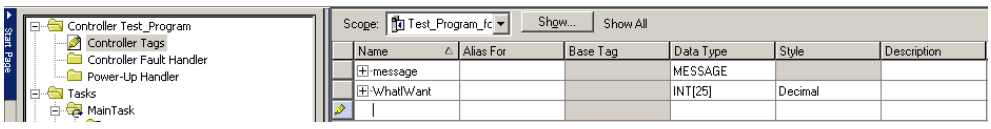

Figure 5: Creating tags for explicit messaging

Step 2:

► Create a NOT that is connected to the MSG command on one line in the program.

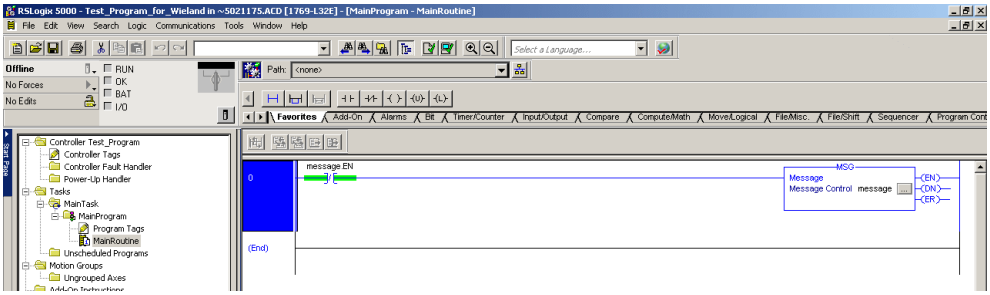

Figure 6: Programming the main routine for explicit messaging

► Select **message.EN** for the NOT symbol. This ensures that the message will be requested repeatedly.

Step 3:

► Edit the MSG command as follows:

- Configuration: Select **CIP Generic**. The **Class** describes the object. **Class 72** stands for Full Data Set Transfer. **Instance 1** and **Attribute 1** determine the data type (in this case **Data Set 1**). The tag **WhatIWant** must be chosen as **Destination**.

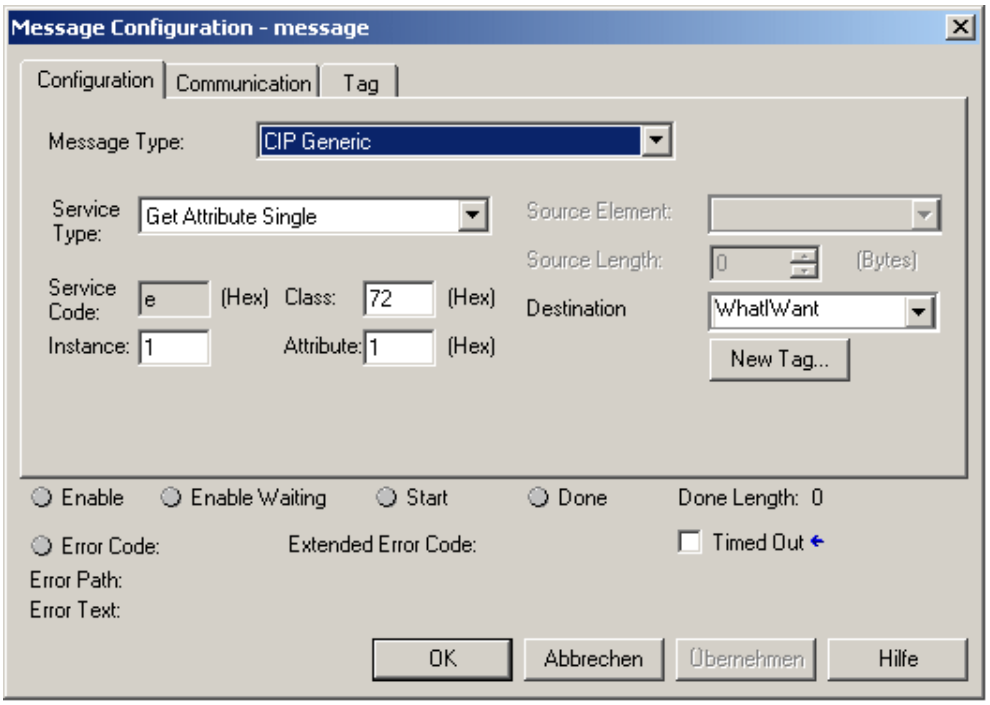

Figure 7: Explicit messaging – message configuration

- Communication: The message must contain the **Path** to the gateway. In this example, the path is 10.4.209.51.

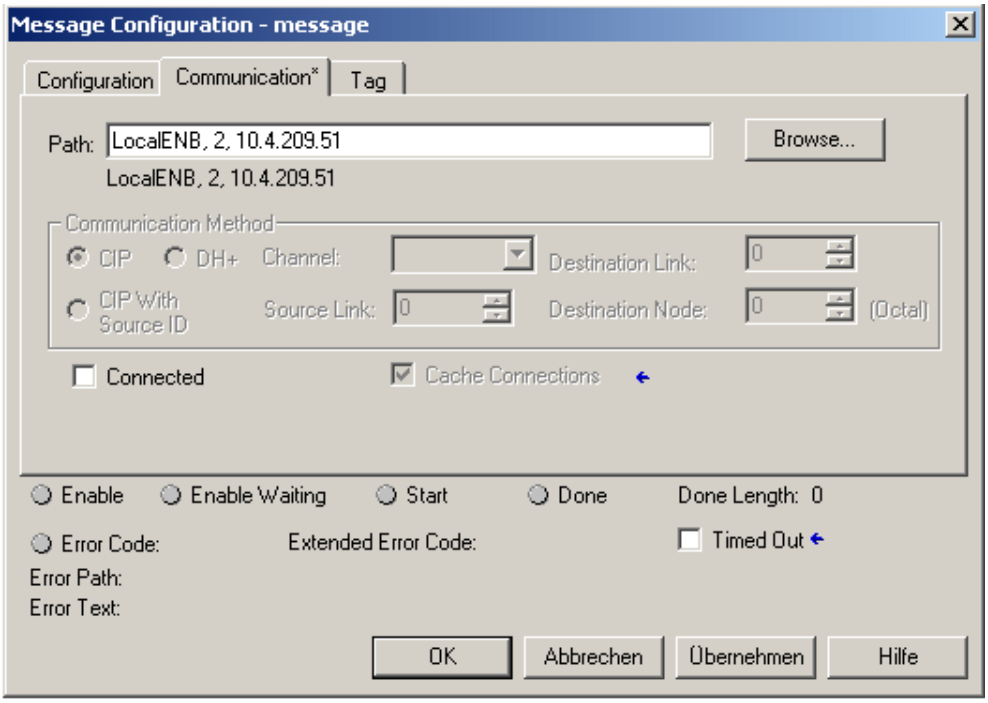

Figure 8: Explicit messaging – configuring communication

4.1.2 FX0-GMOD Modbus TCP gateway

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

4.1.2.1 Basic configuration – Assigning device name and IP address

Procedure

1. Start the configuration software and load the hardware configuration, including the Modbus® TCP gateway.
2. Click on the **Interfaces** button above the main window and select the FX0-GMOD or double-click on the FX0-GMOD in the **Hardware configuration** to open the dialog box for gateway configuration.
3. Click on **Gateway configuration**. The following dialog box is displayed:

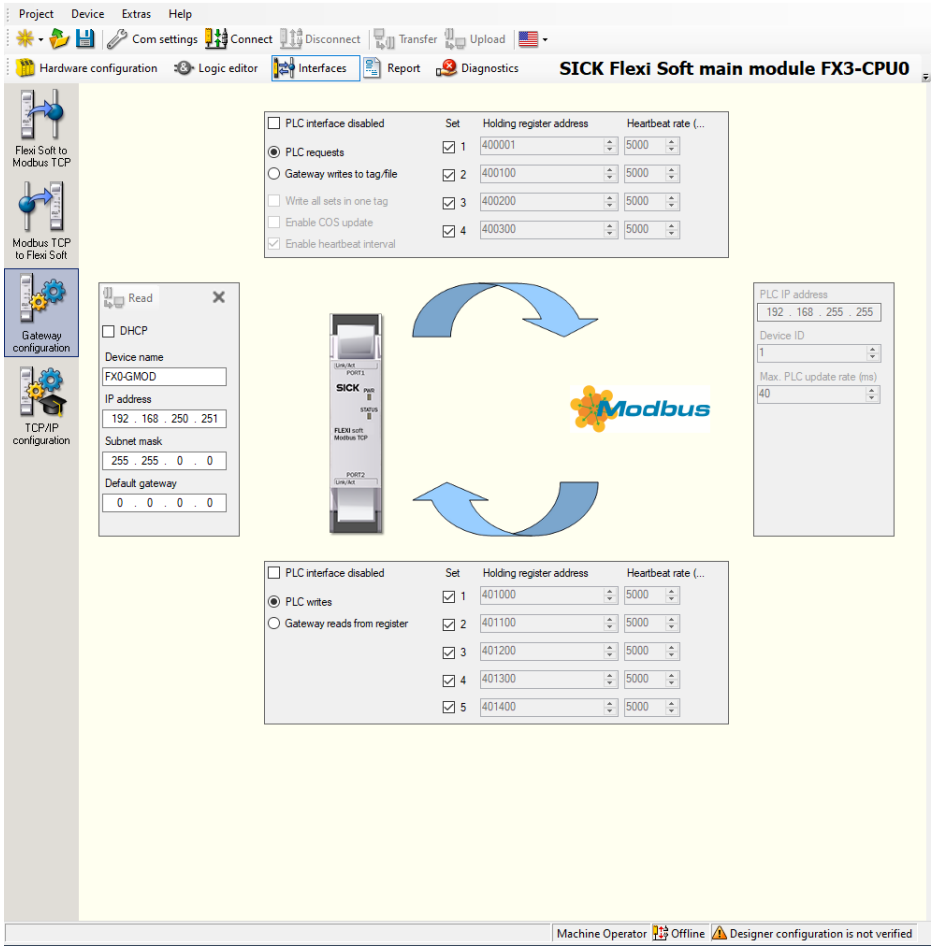

Figure 9: Configuration window for the Modbus® TCP gateway

4. If you wish, change the device name of the gateway.
5. Enter an IP address for the gateway and, if necessary, a subnet mask and an IP address for a default gateway.
6. Click on **Connect** to switch to online mode.
7. Click on **Transfer** to transfer the configuration to the Flexi Soft system.

4.1.2.2 Configuring the Modbus TCP interface

Properties of the Flexi Soft Modbus® TCP gateway

- Modbus TCP with client and server operation
- Supports the standard addressing conventions for Modbus® TCP
- Ethernet TCP/IP socket interface, polling, and auto-update function
- With a firmware version ≥ V2.01.0: Data can also be read out as the Word data type.

Requests to the PLC for Modbus TCP

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

The configuration steps in this section determine the way in which the data is transmitted to the higher-level controller. In general, there are two different transmission types (as a server and as a client) for both directions, i.e., for Flexi Soft to the network and network to Flexi Soft:

- Server reception type – **PLC requests** (polling/gateway as a server)

This reception type makes it possible for the PLC to request data regularly using polling. With this reception type, the data is returned in the response to the message requesting the data. The PLC requests data by accessing the reception data address of the FX0-GMOD module with a **Read Holding Registers** telegram.

- Client reception type – **Gateway writes to tag/file** (auto update, gateway as client)
  If the data received by the FX0-GMOD via the backplane interface is to be sent to the PLC, this data is then immediately written to a memory location in the PLC.
- Server transmission type – **PLC writes** (gateway as server)
  With this transmission type, the PLC sends telegrams to the FX0-GMOD in order to write to the output data sets. For this purpose, the PLC writes the data to defined addresses.
- Client transmission type – **Gateway reads from register** (auto update, gateway as client)
  With this transmission type, the FX0-GMOD polls the PLC for the output data sets.

NOTE
The configuration is incorrect if the IP address of the PLC is 0.0.0.0 and the gateway for one or both of the transmission directions is configured as client.

Number of possible connections

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

Table 31: Number of possible connections

Transmission type	Maximum number of connections
Rx (to PLC/remote peer): client Tx (from PLC/remote peer): client	V2: Rx and Tx: 1 V3: Rx and Tx: 1
Rx (to PLC/remote peer): client Tx (from PLC/remote peer): server	V2: Rx: 1 and Tx: 31 V3: Rx: 1 and Tx: 12
Rx (to PLC/remote peer): server Tx (from PLC/remote peer): client	V2: Rx: 31 and Tx: 1 V3: Rx: 11 and Tx: 1
Rx (to PLC/remote peer): server Tx (from PLC/remote peer): server	V2: Rx and Tx: 32 V3: Rx and Tx: 16

4.1.2.3 Client mode

Overview

With this transmission type, the gateway functions as a client.

Configuration

Table 32: Configuration guideline – gateway as client

Required settings in the gateway configuration	Necessary settings in the PLC program and/or in the Modbus TCP configuration device
Select **Gateway writes to tag/file** and/or **Gateway reads from register** to configure the gateway as a client.	–
Select which data is to be written to the PLC or is to be read from it.	–

Required settings in the gateway configuration	Necessary settings in the PLC program and/or in the Modbus TCP configuration device
Determine the location in the PLC memory to which the selected data is to be written: Enter register address(es). Example: “400001” Determine the location in the PLC memory from which the selected data is to be read: Enter register addresses.	Make sure that the register addresses entered on the gateway are available and contain the data specified for the Flexi Soft system. **Note**: The Modbus TCP communication uses port 502 as default.
Determine how often this data is to be transmitted.	–
Determine the location in the Modbus TCP network from which the data is to be read and to which it is to be written: Enter the IP address and device ID.	–

NOTE
Numbering of the register addresses starts at 1, but Modbus addressing starts at 0. 1 must therefore be subtracted from the register address to ensure correct addressing. For example, the current system time (register 4200) is sent in the telegram as Modbus start address 4199.

4.1.2.3.1 Configuring the gateway as a client

Prerequisites

- The register addresses of the input data sets and the output data sets (set in the configuration software) must be the same as defined in the PLC.
- The variables in the PLC that will store the data or from which the data will be requested must meet the following conditions:
  - They must be in the address range 40xxxx (for Schneider Modicon PLC).
  - They must be an array of 16-bit words.
  - They must be long enough to hold the input data set array and the entire output data set.
- All input data sets and output data sets are transmitted in integer format (16-bit word), whereby the first byte needs to be placed in the least significant or rightmost byte of the integer, and the second byte needs to be placed in the most significant or leftmost byte of the integer.

Procedure

1. Start the configuration software and load the hardware configuration, including the Modbus® TCP gateway.
2. Click on the **Interfaces** button above the main window and select the FX0-GMOD or double-click on the FX0-GMOD in the **Hardware configuration** to open the dialog box for gateway configuration.
3. Click on **Gateway configuration**. The following dialog box is displayed:

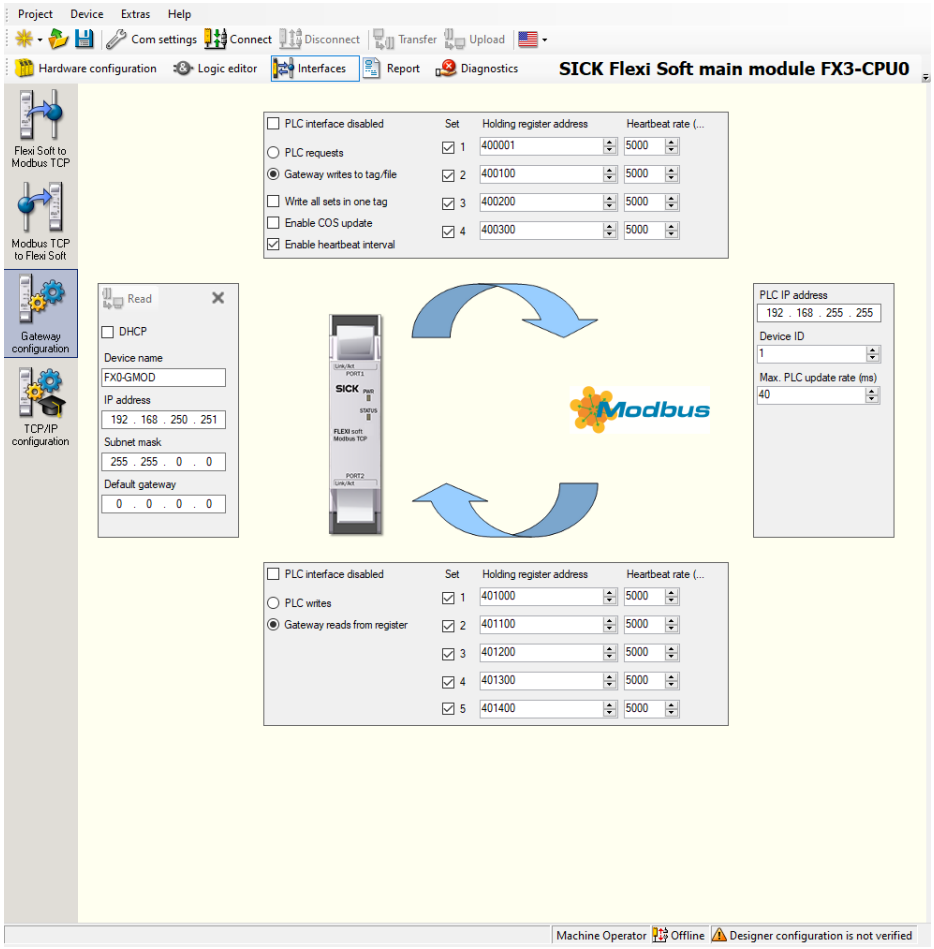

Figure 10: Configuration of the Modbus® TCP gateway as a client

4. Select the transmission type in the **Gateway configuration** dialog box: Select **Gateway writes to tag/file** for the Flexi Soft to network direction. Select **Gateway reads from register** for the network to Flexi Soft direction.
5. Select the checkboxes for the required data sets in the relevant configuration area to select the data that is to be written to the PLC or read from it: see "Data transferred to the network (network input data sets)", page 128).
6. Specify the location in the Modbus TCP network to which or from which the selected data is to be written or read. Enter addresses in the **Holding register address** input field (max. 20 characters).
7. For the Flexi Soft to network transmission direction, select the **All data sets in one tag** option if all data sets are to be written to the PLC memory in a single tag.
8. For the Flexi Soft to network transmission direction, specify how often the data is to be transferred.
   - Select the **Enable COS update** (update in the event of change of state) option if the FX0-GMOD must update the data in the PLC immediately if something in the data sets changes.
   - Select the **Enable heartbeat rate** option to update the selected data sets cyclically with the set **Heartbeat rate** in milliseconds.
   - Both checkboxes can be selected at the same time.
9. For the network to Flexi Soft transmission direction, specify how often the data is to be read.
   - Enter a **Heartbeat rate** to update the selected data sets at the specified interval.

10. Specify the location from which the selected data is to be written in the Modbus® TCP network or the location from which it is to be read. Enter the IP address and Modbus® device ID of the PLC.
11. The **Max. PLC update rate** determines the maximum rate for transferring data sets to or from the PLC. The setting is dependent on the processing speed of the PLC and can be between 10 ms and 65,535 ms. The default setting of 40 ms is suitable for most PLCs.

NOTE

If the value entered for the Max. PLC update rate is higher than the Heartbeat rate set for writing to or reading from the PLC, then the heartbeat rate is automatically increased to this value (i.e., slowed down).

12. Click on **Connect** to switch to online mode.
13. Click on **Transfer** to transfer the configuration to the Flexi Soft system.

4.1.2.4 Server mode

Overview

With this transmission type, the gateway functions as a server. It sends data to the PLC on request and the PLC writes the data to the gateway.

Configuration

Table 33: Configuration guideline – gateway as server

Required settings in the gateway configuration	Necessary settings in the PLC program and/or in the Modbus TCP configuration device
Select **PLC requests** and **PLC writes** in the gateway configuration window.	–
–	Determine which data is to be written to the gateway or is to be read from it. Make sure that the PLC program writes the data to the addresses defined for the gateway ("figure X").

Data addressing

NOTE

- All data sets can only be read or written as a complete block. It is not possible to read or write individual bits or bytes.
- If the control system cannot be configured in words, the data scope must be converted accordingly.

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

Table 34: Data addressing for the FX0-GMOD as server (unit ID: 1)

Address (base 1)	Description	Access type	Scope [words]
1000	Request data from all activated input data sets	Read	16 ... 101 [1]
1100	Request data from input data set 1	Read	25
1200	Request data from input data set 2	Read	16
1300	Request data from input data set 3	Read	30
1400	Request data from input data set 4	Read	30
2000	Write all activated output data sets	Write	5 ... 25 [2]
2100	Write data from output data set 1	Write	5

Address (base 1)	Description	Access type	Scope [words]
2200	Write data from output data set 2	Write	5
2300	Write data from output data set 3	Write	5
2400	Write data from output data set 4	Write	5
2500	Write data from output data set 5	Write	5

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

4.1.2.4.1 Configuring the gateway as a server

Prerequisites

- The device ID must be “1”.
- Output data set: The telegram must be sent in the word format.
- Input data set: The variable in the PLC that is to receive the data must satisfy the following conditions:
  - It must be in the address range 40xxxx (for Schneider Modicon PLC).
  - It must be an array of 16-bit words.
  - It must be long enough to hold the data set array(s).
- All input data sets and output data sets are transmitted in integer format (16-bit word), whereby the first byte needs to be placed in the least significant or right-most byte of the integer, and the second byte needs to be placed in the most significant or leftmost byte of the integer.

Procedure

1. Start the configuration software and load the hardware configuration, including the Modbus® TCP gateway.
2. Click on the **Interfaces** button above the main window and select the FX0-GMOD or double-click on the FX0-GMOD in the **Hardware configuration** to open the dialog box for gateway configuration.
3. Click on **Gateway configuration**. The following dialog box is displayed:

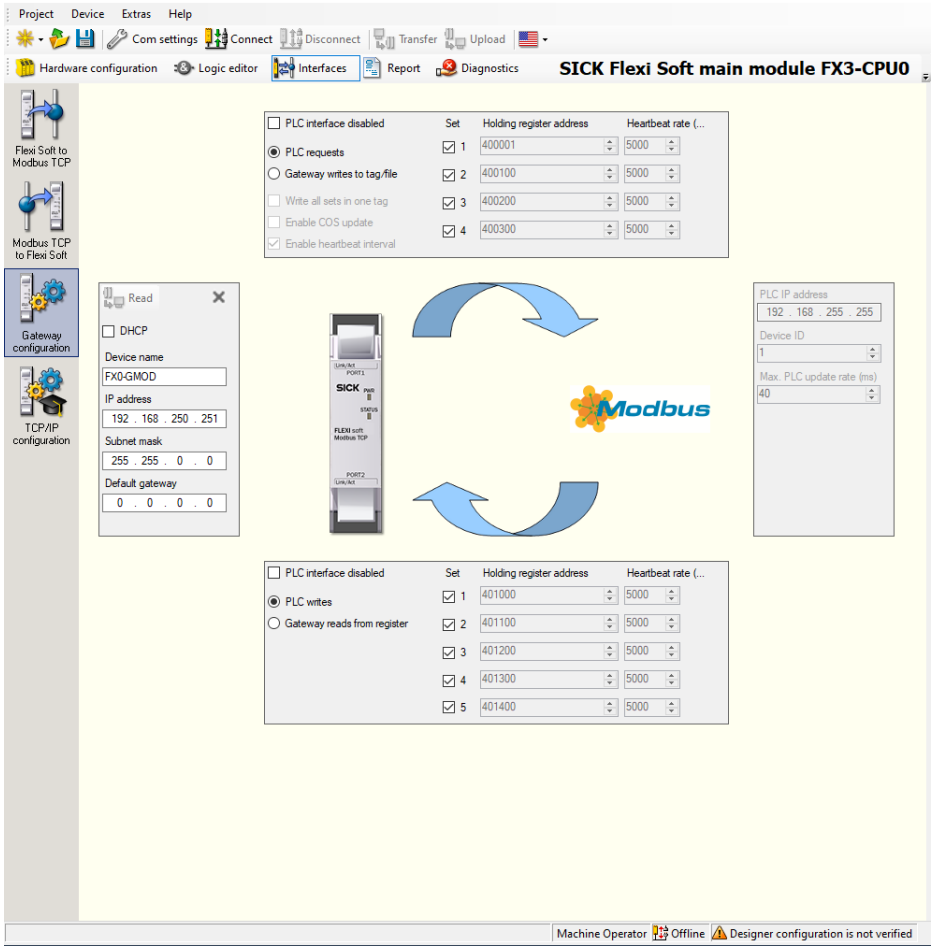

Figure 11: Configuration of the Modbus® TCP gateway as a server

4. Select the transmission type in the dialog box: Select the **PLC requests** option for the Flexi Soft to network direction. Select the **PLC writes** option for the network to Flexi Soft direction.
5. Select the checkboxes for the required data sets in the relevant configuration area to select the data that is to be requested or written by the PLC (for a description of the data sets, see "Data transferred to the network (network input data sets)", page 128).
6. Click on **Connect** to switch to online mode.
7. Click on **Transfer** to transfer the configuration to the Flexi Soft system.

4.1.2.5 Commands and error messages

Commands

Table 35: Modbus commands

Modbus command	Value
Read Holding Registers	3
Write Multiple Registers	16 (10h)
Read/Write Multiple Registers	23 (17h)

Error messages

Table 36: Modbus error messages

Fault number	Error message	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, e.g., insufficient data requested for a data set
10	The gateway path is not available.	Invalid configuration, e.g., polling or setting the digital outputs via PLC during operation of the FX0-GMOD in master mode

4.1.3 FX0-GPNT PROFINET IO gateway

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

The FX0-GPNT supports

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

As of firmware version ≥ V3.00.0, the FX0-GPNT also supports

- PROFINET IO, conformance class B
- Netload class 1
- MRP client

4.1.3.1 Basic configuration – Assigning device name and IP address

Configuration and diagnostics of the FX0-GPNT are possible both with the Flexi Soft configuration software and with the PROFINET IO network configuration tool (e.g. SIEMENS SIMATIC Manager).

Configuration via PROFINET IO

In the delivery configuration, a MAC address and a symbolic name are stored in each PROFINET IO field device such as the FX0-GPNT.

NOTE

- The symbolic name of the gateway is **FX0-GPNT**.
- This name is used by the I/O controller (e.g. the PLC) to assign an IP address to the field device.
- If the PLC changes the IP address of the FX0-GPNT, every other Ethernet communication such as a TCP/IP or Ethernet configuration connection which also uses the IP address of the FX0-GPNT is interrupted.

An IP address is assigned in two steps:

► Assign the gateway a unique plant-specific name either using the network configuration tool, e.g. SIEMENS SIMATIC Manager, or using the Flexi Soft configuration software.

► Using the unique plant-specific name, the I/O-Controller (i.e. the PLC) can assign an IP address to the gateway before the system is booted.

NOTE

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

Assigning device name via Flexi Soft Designer

- ► Start Flexi Soft Designer and load the hardware configuration, including the PROFINET IO gateway.
- ► Click on the **Interfaces** button above the main window and select the FX0-GPNT or double-click on the FX0-GPNT in the **Hardware configuration** to open the dialog box for gateway configuration.
- ► Click on the **Gateway configuration** button. The following dialog box is displayed:

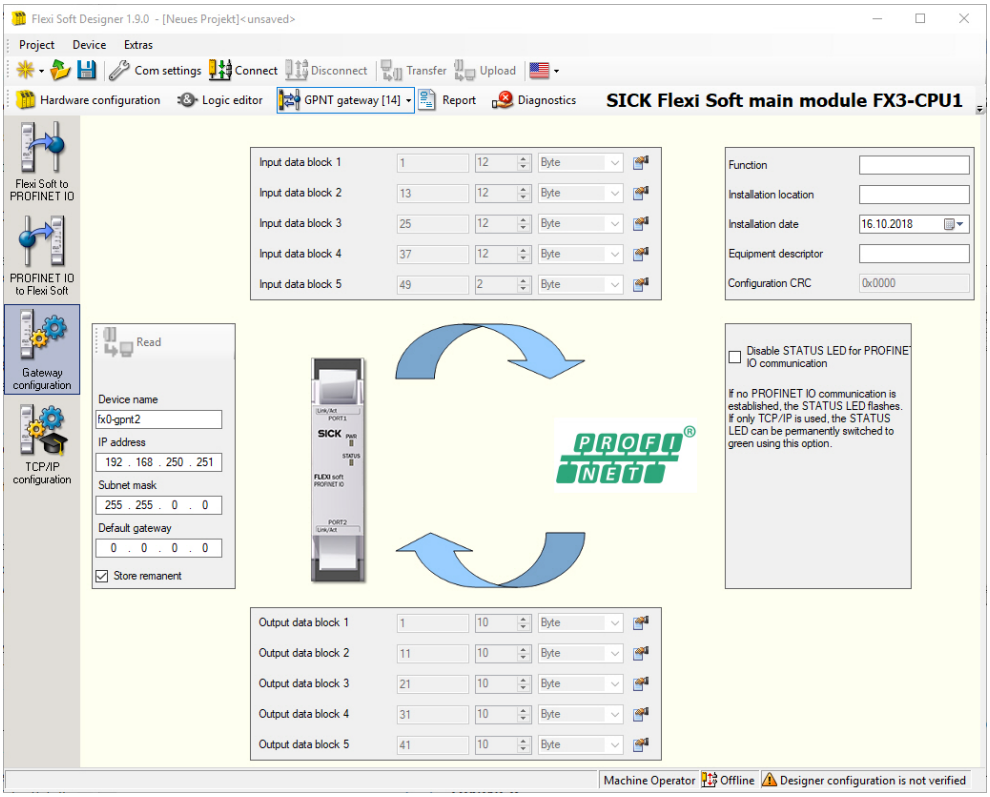

Figure 12: Configuration window for the PROFINET IO gateway

- ► Enter the device name in the **Device name** field.
- ► Click on **Connect** to switch to online mode.
- ► Click on **Transfer** to transfer the configuration to the Flexi Soft system.

NOTE

- The format you use for the device name must correspond to the specifications of the PROFINET IO standard.
- Ensure that the address for the default gateway corresponds to that set for the gateway by the PLC. If no router is used, then SIEMENS SIMATIC STEP 7 uses the same IP address for the default gateway as for the FX0-GPNT.

Assigning IP address via Flexi Soft Designer

The IP address is usually assigned by the PROFINET IO controller (e.g. PLC). However, the FX0-GPNT also allows configuration of the entire Flexi Soft system over Ethernet TCP/IP. In this case, it may be necessary to already assign an IP address to the gateway before setting up the PROFINET IO network. This can also be done via the configuration page (see figure 12, page 44).

Saving the connection data retentively

As of firmware version ≥ V3.00.0, connection data can be saved in the FX0-GPNT retentively. Each time it is restarted, the gateway resets to the most recent connection data that was saved retentively.

Save the connection data retentively.

- ► Check the **Save retentively** box.
- ► Click on **Connect** to switch to online mode.
- ► Click on **Transfer** to transfer the configuration to the Flexi Soft system.

Deactivating the STATUS LED for PROFINET IO communication

If the firmware version is ≥ V2.00.0, you can use the Flexi Soft Designer to stop the STATUS LED from flashing red/green. Otherwise, the LED will flash constantly in the absence of PROFINET IO communication (e.g. if the gateway is being used purely for TCP/IP communication).

- ► Click on the **Interfaces** button above the main window and select the FX0-GPNT or double-click on the FX0-GPNT in the **Hardware configuration** to open the dialog box for gateway configuration.
- ► Click on the **Gateway configuration** button. On the right side of the window, you will find the following configuration area:

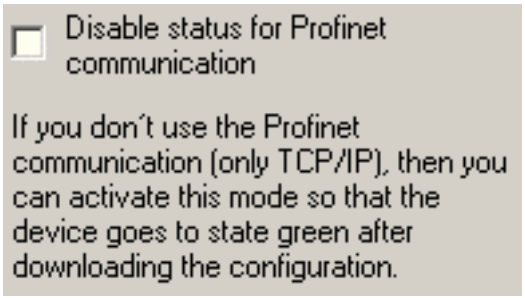

Figure 13: Deactivating the STATUS LED of the FX0-GPNT

- ► Check the box and transfer the configuration. Then, the LED will light up steady ● green, even if there is no PROFINET IO communication in place.

NOTE
This function is available only with firmware ≥ V2.00.0 and Flexi Soft Designer Version ≥ V1.4.0.

4.1.3.2 PROFINET IO configuration – Data selection

Cyclic data

The physical Flexi Soft I/O modules are not represented as typical hardware modules in the PROFINET IO hardware catalog. Instead, the data provided by the Flexi Soft system is split into 10 different data blocks. Each data block represents one “Hardware” module in the PROFINET IO hardware catalog. The GSDML of the Flexi Soft PROFINET IO gateway supports 10 slots, into which the modules can be placed, see figure 14, page 47.

Process data from the Flexi Soft system to the PLC

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

NOTE

- Depending on what PLC you are using, further modules may be displayed (e.g. “universal module”). These modules are not required and should be ignored.
- Input data blocks 1 to 4 each contain 12 bytes while input data block 5 contains 2 bytes.
- The content of the input data sets can be changed using the configuration software. The default content on delivery is as follows:

Table 37: Preset content of input data blocks 1 to 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	Direct gateway output values 1	Not assigned	
Byte 5	Input values, module 6	Output values, module 6	Direct gateway output values 2	Not assigned	
Byte 6	Input values, module 7	Output values, module 7	Direct gateway output values 3	Not assigned	
Byte 7	Input values, module 8	Output values, module 8	Direct gateway output values 4	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

Information on the content of the process image: see "Data transferred to the network (network input data sets)", page 128.

Data from the PLC to the Flexi Soft system

There are five output data blocks with 10 bytes each.

The content of these data blocks can be used as input in the logic editor of the main module or can be routed into another network using a second gateway. A tag name must be assigned to each bit that is to be used so that the desired bits are available in the logic editor or for routing. Bits without a tag name are not available.

Settings in the PROFINET IO network configuration tool

► The FX0-GPNT is shown in the configuration table of SIEMENS SIMATIC Manager – HW Config. Drag the data blocks from the hardware catalog of SIEMENS SIMATIC Manager – HW Config under >>**PROFINET IO** > **Additional field devices** > **Gateway** > **SICK** > **Flexi Soft** > **Data blocks** into the slots of the FX0-GPNT.

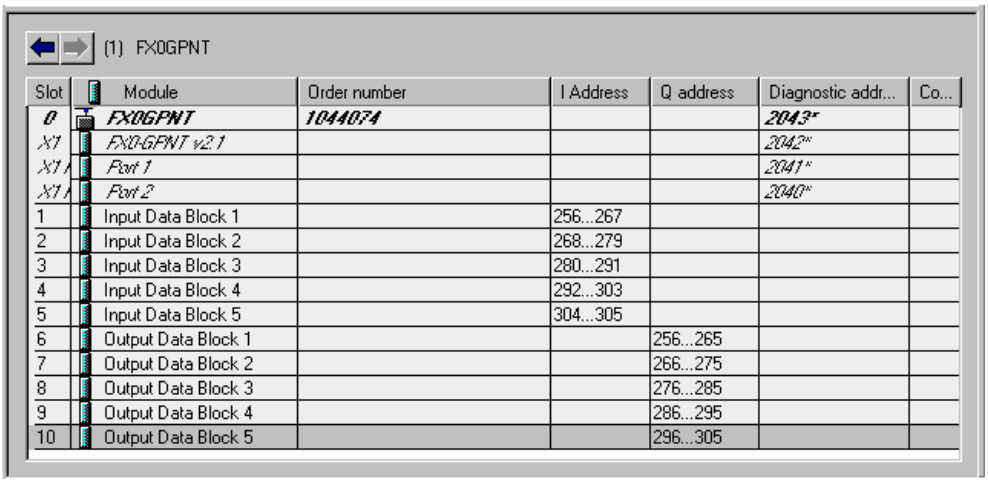

Figure 14: Projecting the FX0-GPNT

NOTE
The **I address** and the **Q address** specify where the cyclic data is available in the memory.

Acyclic data and alarms

Read-out data

The PLC can read out diagnostic data of the Flexi Soft system. The diagnostic information is made available in three data sets, data sets 2, 3 and 4:

- Data set 2 contains the Flexi Soft checksums.
- Data set 3 contains the status of the individual modules with four bytes per module in each case.
- Data set 4 is currently filled with reserved values.

The table below shows the format of the data sets.

In order to access the acyclic data sets, the data must be read out at the corresponding address, as shown in the following table.

Table 38: Memory address for data sets 2, 3 and 4

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

NOTE
Data set 1 is mapped into the cyclically transferred PROFINET IO modules of the device. The content can be defined by the user.

Table 39: Default content of input data sets 2, 3 and 4 of the FX0-GPNT

<table>
<tr><th></th><th>Data set 2</th><th>Data set 3</th><th>Data set 4</th></tr>
<tr><td>Byte 0</td><td rowspan="4">Overall checksum</td><td rowspan="4">Status Module 0</td><td rowspan="60">Reserved</td></tr>
<tr><td>Byte 1</td></tr>
<tr><td>Byte 2</td></tr>
<tr><td>Byte 3</td></tr>
<tr><td>Byte 4</td><td rowspan="4">Flexi Soft checksum</td><td rowspan="4">Status Module 1</td></tr>
<tr><td>Byte 5</td></tr>
<tr><td>Byte 6</td></tr>
<tr><td>Byte 7</td></tr>
<tr><td>Byte 8</td><td rowspan="4">FX3-CPU0 and FX3-CPU1: reserved
FX3-CPU2 and FX3-CPU3: ACR checksum</td><td rowspan="4">Status Module 2</td></tr>
<tr><td>Byte 9</td></tr>
<tr><td>Byte 10</td></tr>
<tr><td>Byte 11</td></tr>
<tr><td>Byte 12</td><td rowspan="20">Reserved</td><td rowspan="4">Status Module 3</td></tr>
<tr><td>Byte 13</td></tr>
<tr><td>Byte 14</td></tr>
<tr><td>Byte 15</td></tr>
<tr><td>Byte 16</td><td rowspan="4">Status Module 4</td></tr>
<tr><td>Byte 17</td></tr>
<tr><td>Byte 18</td></tr>
<tr><td>Byte 19</td></tr>
<tr><td>Byte 20</td><td rowspan="4">Status Module 5</td></tr>
<tr><td>Byte 21</td></tr>
<tr><td>Byte 22</td></tr>
<tr><td>Byte 23</td></tr>
<tr><td>Byte 24</td><td rowspan="4">Status Module 6</td></tr>
<tr><td>Byte 25</td></tr>
<tr><td>Byte 26</td></tr>
<tr><td>Byte 27</td></tr>
<tr><td>Byte 28</td><td rowspan="4">Status Module 7</td></tr>
<tr><td>Byte 29</td></tr>
<tr><td>Byte 30</td></tr>
<tr><td>Byte 31</td></tr>
<tr><td>Byte ...</td><td rowspan="6">Not available</td><td>...</td></tr>
<tr><td>Byte 56</td><td rowspan="5">Status Module 14
Modules 13 and 14 are always the gateways.</td></tr>
<tr><td>Byte 57</td></tr>
<tr><td>Byte 58</td></tr>
<tr><td>Byte 59</td></tr>
<tr><td>Length</td><td>32 bytes</td><td>60 bytes</td><td>60 bytes</td></tr>
</table>

Interpretation of the module status bit in data set 3: see "Error and status information of the modules", page 133

I&M information

The FX0-GPNT supports the I&M information defined in the PROFINET IO specification. The following I&M information can be read out of the device:

Table 40: I&M information of the FX0-GPNT

I&M field	Size	Value
Manufacturer ID	2 bytes	257
Order ID	20 bytes	"1044074 " (must be padded with 13 spaces)
Serial number	16 bytes	Read from I^2C
Hardware revision	4 bytes	Read from I^2C
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
I&M version	2 bytes	1.1
I&M supported	2 bytes	0

With a firmware version ≥ V3.00.0, the following I&M information can be displayed and edited using Flexi Soft Designer:

- Function
- Installation site
- Installation date
- Equipment descriptor
- Configuration CRC

Editing the I&M information

► Switch to the **Gateway configuration**.
► Change the I&M information as required.
► Click on **Connect** to switch to online mode.
► Click on **Transfer** to transfer the configuration to the Flexi Soft system.

Alarms

Alarms can be read acyclically using the PROFINET IO alarm infrastructure. If an error occurs in one of the Flexi Soft modules, the PROFINET IO gateway sends the corresponding diagnostic alarm into the network. The error LEDs of the PLC light up as a result; the details of the diagnostic alarm (text and help) are then available via the SIMATIC PLC interface. The function block RALRM (SFB54) in OB82 (diagnostic interrupt) allows the details of the sent alarm to be made available directly in the PLC program.

NOTE

- All alarms are output to module 0.
- The subslot number indicates the Flexi Soft module that caused the alarm. Number 0 = main module, 1 = 1st expansion module, 2 = 2nd expansion module... 13 = 1st gateway, 14 = 2nd gateway.
- The cause of the alarm is indicated by an error message from the GSDML file. Up to 32 different error messages are possible for each Flexi Soft module type.
- The same diagnostic information can also be read out from data set 3.

The following table shows the PROFINET IO error types (as defined in the GSDML file) and the corresponding error messages.

Table 41: PROFINET IO error types

Error type	Message	
	Cause oferror [1]	Error definition
0100	Main module	Reserved
0101		Internal error: internal tests failed.
0102		External error
0103		Reserved
0104		Configuration status of a module in the system is incompatible or invalid.
0105		Voltage supply outside of the specified range
0106		EFI1 communication error
0107		EFI2 communication error
0108 ... 0131		Reserved

Error type	Message	
	Cause oferror [1]	Error definition
0200	FX3-XTIO/ FX3-XTDI	Reserved
0201		Internal error: internal tests failed.
0202		External error
0203		Reserved
0204		Configuration is incompatible or invalid.
0205		Voltage supply outside of the specified range
0206 ... 0207		Reserved
0208		Input 1–2 dual-channel evaluation: error
0209		Input 3–4 dual-channel evaluation: error
0210		Input 5–6 dual-channel evaluation: error
0211		Input 7–8 dual-channel evaluation: error
0212 ... 0215		Reserved
0216		Input 1 external error for test signal. Check for stuck-at-high or cabling error.
0217		Input 2 external error for test signal. Check for stuck-at-high or cabling error.
0218		Input 3 external error for test signal. Check for stuck-at-high or cabling error.
0219		Input 4 external error for test signal. Check for stuck-at-high or cabling error.
0220		Input 5 external error for test signal. Check for stuck-at-high or cabling error.
0221		Input 6 external error for test signal. Check for stuck-at-high or cabling error.
0222		Input 7 external error for test signal. Check for stuck-at-high or cabling error.
0223		Input 8 external error for test signal. Check for stuck-at-high or cabling error.
0224		Output 1 test evaluation stuck-at-high error
0225		Output 1 test evaluation stuck-at-low error
0226		Output 2 test evaluation stuck-at-high error
0227		Output 2 test evaluation stuck-at-low error
0228		Output 3 test evaluation stuck-at-high error
0229		Output 3 test evaluation stuck-at-low error
0230		Output 4 test evaluation stuck-at-high error
0231		Output 4 test evaluation stuck-at-low error
0300	PROFIBUS-DP gateway	Reserved
0301		Internal error: internal tests failed.
0302 ... 0303		Reserved
0304		Configuration is incompatible or invalid.
0305 ... 0331		Reserved

Error type	Message	
	Cause oferror [1]	Error definition
0400	CANopen gateway	Reserved
0401		Internal error: internal tests failed.
0402 ... 0403		Reserved
0404		Configuration is incompatible or invalid.
0405 ... 0431		Reserved
0500	DeviceNet gateway	Reserved
0501		Internal error: internal tests failed.
0502 ... 0503		Reserved
0504		Configuration is incompatible or invalid.
0505 ... 0531		Reserved
0600	Modbus TCP gateway	Reserved
0601		Internal error: internal tests failed.
0602 ... 0603		Reserved
0604		Configuration is incompatible or invalid.
0605 ... 0631		Reserved
0700	EtherNet/IP™ gateway	Reserved
0701		Internal error: internal tests failed.
0702 ... 0703		Reserved
0704		Configuration is incompatible or invalid.
0705 ... 0731		Reserved
0800	PROFINET IO gateway	Reserved
0801		Internal error: internal tests failed.
0802 ... 0803		Reserved
0804		Configuration is incompatible or invalid.
0805 ... 0831		Reserved
1200	CC-Link gateway	Reserved
1201		Internal error: internal tests failed.
1202 ... 1203		Reserved
1204		Configuration is incompatible or invalid.
1205 ... 1231		Reserved

Error type	Message	
	Cause oferror [1]	Error definition
1500	Sercos III gateway	Reserved
1501		Internal error: internal tests failed.
1502 ... 1503		Reserved
1504		Configuration is incompatible or invalid.
1505 ... 1531		Reserved
1600	EtherCAT gateway	Reserved
1601		Internal error: internal tests failed.
1602 ... 1603		Reserved
1604		Configuration is incompatible or invalid.
1605 ... 1631		Reserved
1900	EFI-pro gateway	Reserved
1901		Internal error: Internal tests have failed.
1902 ... 1903		Reserved
1904		Configuration is incompatible or invalid.
1905 ... 1931		Reserved
2000 ... 3131	Other gateways	Reserved
3200	FX0-STIO	Reserved
3201		Internal error: internal tests failed.
3202 ... 3203		Reserved
3204		Configuration is incompatible or invalid.
3205		Voltage supply outside of the specified range
3206		Reserved
3207		Output load (overcurrent) monitoring
3208 ... 3231		Reserved

Error type	Message	
	Cause oferror [1]	Error definition
3300	FX3-MOCx	Reserved
3301		Internal error: internal tests failed.
3302 ... 3303		Reserved
3304		Configuration is incompatible or invalid.
3305		Reserved
3306		Encoder 1 status
3307		Encoder 2 status
3308		Encoder 1 teach status
3309		Encoder 2 teach status
3310 ... 3311		Reserved
3312		User-defined status bit 1
3313		User-defined status bit 2
3314		User-defined status bit 3
3315		User-defined status bit 4
3316		User-defined monitoring bit 1
3317		User-defined monitoring bit 2
3318		User-defined monitoring bit 3
3319		User-defined monitoring bit 4
3320		User-defined monitoring bit 5
3321		User-defined monitoring bit 6
3322		User-defined monitoring bit 7
3323		User-defined monitoring bit 8
3324		User-defined monitoring bit 9
3325		User-defined monitoring bit 10
3326		User-defined monitoring bit 11
3327		User-defined monitoring bit 12
3328		User-defined monitoring bit 13
3329		User-defined monitoring bit 14
3330		User-defined monitoring bit 15
3331		User-defined monitoring bit 16

Error type	Message	
	Cause oferror [1]	Error definition
3400	FX3-XTDS	Reserved
3401		Internal error: internal tests failed.
3402 ... 3403		Reserved
3404		Configuration is incompatible or invalid.
3405		Voltage supply outside of the specified range
3406		Reserved
3407		Output load (overcurrent) monitoring
3408		Input 1–2 dual-channel evaluation: error detected
3409		Input 3–4 dual-channel evaluation: error detected
3410		Input 5–6 dual-channel evaluation: error detected
3411		Input 7–8 dual-channel evaluation: error detected
3412 ... 3415		Reserved
3416		External test signal Input 1: error
3417		External test signal Input 2: error
3418		External test signal Input 3: error
3419		External test signal Input 4: error
3420		External test signal Input 5: error
3421		External test signal Input 6: error
3422		External test signal Input 7: error
3423		External test signal Input 8: error
3424 ... 3431		Reserved
3500	FX3-ANA0	Reserved
3501		Internal error: internal tests failed.
3502 ... 3503		Reserved
3504		Configuration is incompatible or invalid.
3505 ... 3531		Reserved
3600 ... 6331	Other modules	Reserved

1) The GSDML file contains the error types in decimal notation. The error types are represented as hexadecimal values in data block 4 of the TIA portal.

2) The status of this bit can be defined for a specific application in the FX3-MOCx logic, e.g. in order to indicate prohibited movements of an axis that were detected by an FX3-MOCx function block.

4.1.4 FX0-GETC EtherCAT gateway

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

The FX0-GETC is an EtherCAT slave device. It supports the following services, which are mandatory to enable the full range of functions:

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

4.1.4.1 Installing the gateway in the Flexi Soft system

This section describes the basic steps for installing the gateway in the Flexi Soft system. Further information can be found in the following sections.

Adding the gateway to a Flexi Soft system

► Install the gateway and connect it to the EtherCAT network (see the operating instructions titled “Flexi Soft Gateways Hardware” (SICK part number 8012662)).
► Start Flexi Soft Designer and load the hardware configuration, including the FX0-GETC, or create a new Flexi Soft system with an FX0-GETC in Flexi Soft Designer.

Configuring the gateway in the Flexi Soft system

► Click on the **Interfaces** button above the main window and select the FX0-GETC or double-click on the FX0-GETC in the **Hardware configuration** to open the dialog box for gateway configuration. The configuration window is divided into three areas: **Flexi Soft to EtherCAT**, **EtherCAT to Flexi Soft**, and **Gateway configuration**, which can be accessed via the buttons on the left.
► In the **Flexi Soft to EtherCAT** area, select the data which is to be transferred to the EtherCAT network from the Flexi Soft system. You can use up to 50 bytes, which are split into five input data sets, each with 10 bytes (see "Process image configuration", page 140).
► In the **EtherCAT to Flexi Soft** area, select the data which is to be transferred to the Flexi Soft system from the EtherCAT network. You can use up to 50 bytes, which are split into five output data sets, each with 10 bytes.
► In the **Gateway configuration** area, you can enter the required device name of the gateway in the Flexi Soft system. The preset name of the gateway is “GETC”.

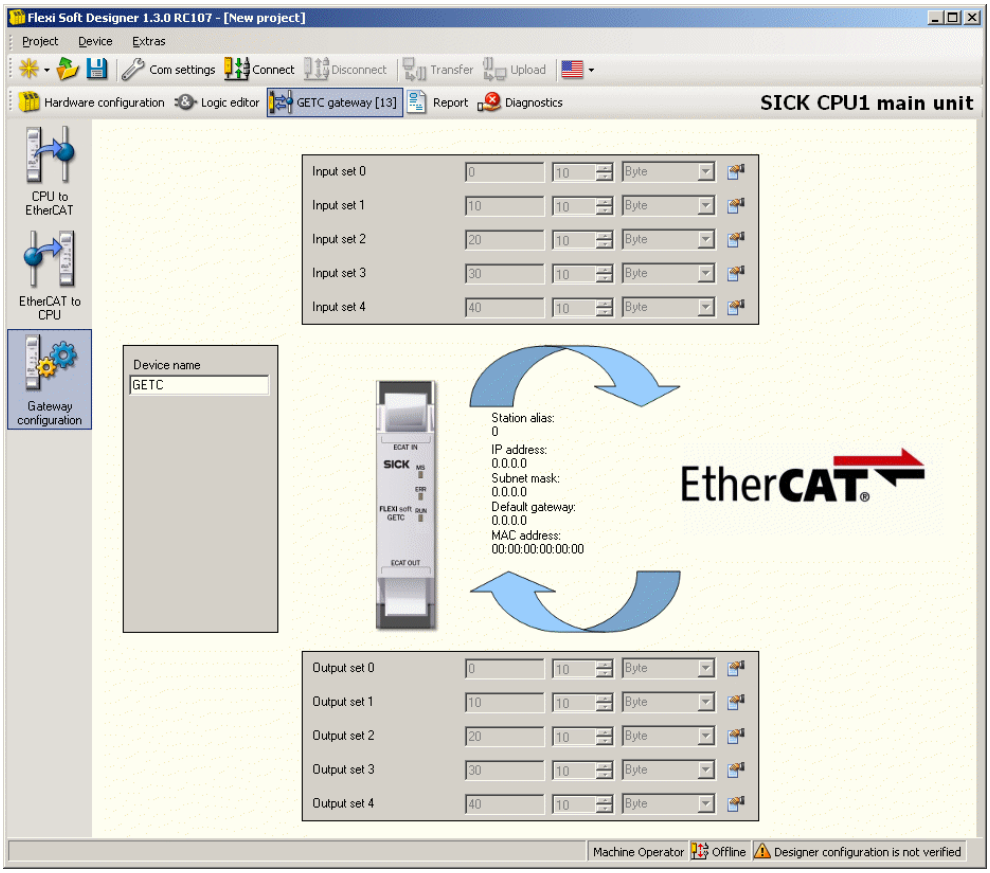

Figure 15: Configuration window for the EtherCAT gateway

NOTE

The second station address (Station Alias) of the gateway and the EoE parameters are displayed in the middle of the screen. This data can be changed only by means of the EtherCAT network configuration tool (e.g. TwinCAT).

4.1.4.2 EtherCAT configuration of the gateway

Important information

NOTE

This documentation does not cover how to set up the EtherCAT network in the network configuration tool. Nor does it deal with the other components of the automation system project within this tool.

Prerequisites

- The EtherCAT project is set up in the configuration software, e.g., TwinCAT.

EtherCAT configuration of the gateway

The steps described below are necessary to configure communication between the PLC and the gateway. The examples shown here relate to configurations that have been created using TwinCAT V2.11.0.

4.1.4.2.1 Step 1: Install the EtherCAT slave information file (ESI)

Overview

The ESI file SICK-FX0-GETC.xml contains the necessary information for integrating the FX0-GETC into the EtherCAT network. Before you can use the FX0-GETC as a device in the EtherCAT network configuration tool (e.g., TwinCAT) for the first time, you must first install the gateway ESI file in the hardware catalog of the tool.

Procedure

1. Download the ESI file from the FX0-GETC product page at www.sick.com. Alternatively, the ESI file can be found in the Flexi Soft configuration software program folder (the default directory is “C:\Program Files\SICK\FlexiSoft\Device-Descriptions\FX0-GETC_ESI”).
2. Follow the instructions for installing ESI files provided by the online help system or the user manual for the EtherCAT network configuration tool.

Example

Installing the ESI file using TwinCAT

1. Copy the ESI file SICK-FX0-GETC.xml to the TwinCAT directory under “TwinCAT\Io\EtherCAT\”.
2. Restart TwinCAT.

✓ The ESI cache is updated.

4.1.4.2.2 Step 2: Add the gateway in the EtherCAT network

Overview

To make the system data of the Flexi Soft system available in the process image of the PLC, you must first add the gateway to the hardware configuration. The procedure for this depends on the configuration software of the PLC you are using and is usually described in the associated manual.

Example

Adding the FX0-GETC using TwinCAT

► To integrate the gateway into the EtherCAT network manually, use the **Add box** command and select the Flexi Soft EtherCAT gateway.

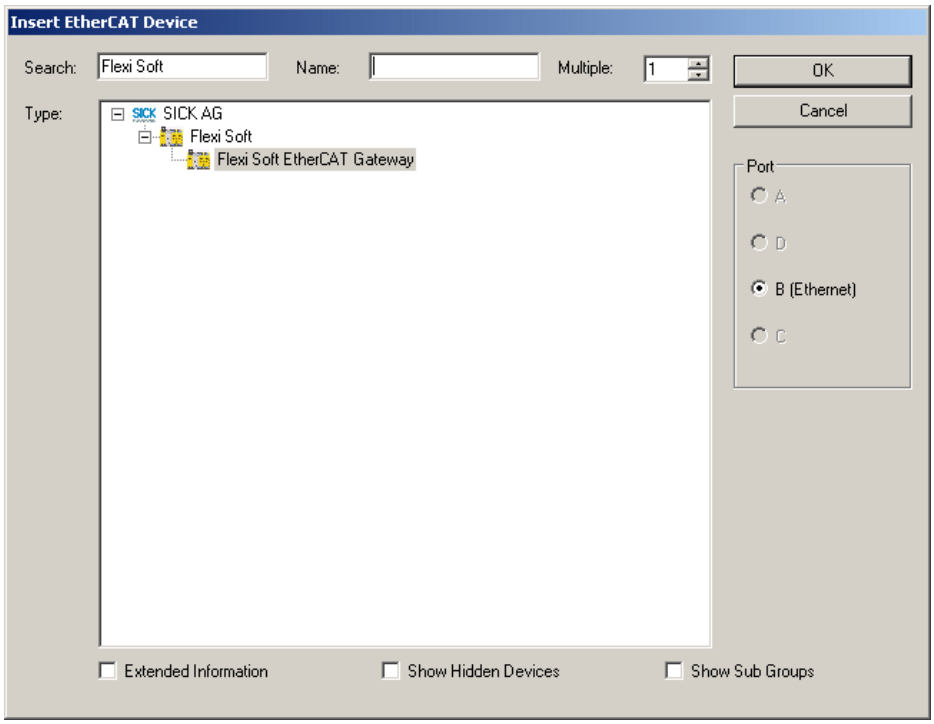

Figure 16: Example for integrating FX0-GETC into an EtherCAT network

► Alternatively, search for devices in the network using the **Scan boxes** command.

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

Overview

The number of records that can be linked to PLC variables depends on the process data objects (PDOs) selected. Once you have added the device to the automation network, you need to select and configure the PDOs that you will be using.

The FX0-GETC provides five input PDOs for transferring input data to a connected PLC. These can be used alternately. In other words, only one of these five input PDOs may be active at one time. There is an input PDO for 10-byte input data (= 1 data set used in the configuration software), one for 20-byte data (= 2 data sets used), etc., up to a maximum of 50 bytes. In accordance with this, one of the five available output PDOs of 10 to 50 bytes needs to be selected to which the PLC output data can be written.

Table 42: Process data objects of the FX0-GETC

Input PDOs		
Index	**Size**	**Content**
1A00h	11 bytes	Diag byte + input record 1
1A01h	21 bytes	Diag byte + input records 1–2
1A02h	31 bytes	Diag byte + input records 1–3
1A03h	41 bytes	Diag byte + input records 1–4
1A04h	51 bytes	Diag byte + input records 1–5
Output PDOs		
Index	**Size**	**Content**
1600h	10 bytes	Output record 1
1601h	20 bytes	Output records 1–2
1602h	30 bytes	Output records 1–3

1603h	40 bytes	Output records 1–4
1604h	50 bytes	Output records 1–5

Important information

NOTE

- The structure of the PDOs is predefined and cannot be changed.
- The input PDOs contain an additional first byte for the diagnostics flag (diag). This byte is set to True (“1”) if a new diagnostic message (CoE object 10F3h) is available, and set to False (“0”) if all diagnostic messages have been acknowledged.
- Only one input PDO and one output PDO can be active at any one time.
- If the selected PDO is larger than the configured process data, the unused data ranges are filled with nulls.
- If the selected PDO is smaller than the configured process data, the excess data are truncated.

Procedure

► For each transmission direction (input and output), select one of the five available PDOs with a suitable size for the process data used in the EtherCAT network configuration tool.

Example

Selecting the PDOs using TwinCAT

1. On the **Process data** tab, select the desired PDO type (**Inputs** or **Outputs**) in the **Sync Manager** selection list.
2. Next select the desired PDO in the **PDO assignment** selection list. To select a different PDO, the active PDO needs to be deactivated first.
3. In the **Download** area, select the **PDO Assignment** option.
4. Do not select **PDO configuration**, as the PDO configuration is predefined and cannot be modified.

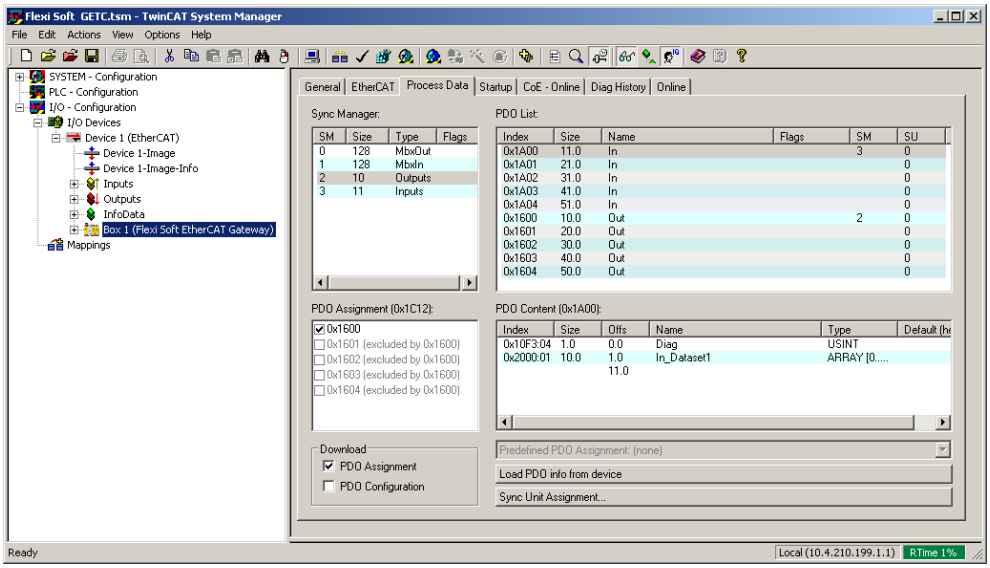

Figure 17: PDO configuration in the EtherCAT network configuration tool

4.1.4.3 Input data – Flexi Soft to EtherCAT

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

NOTE

- Each input data set contains 10 bytes.
- The default content of the input data sets can be changed using the configuration software.
- If an input data set contains data, all 10 bytes are sent over EtherCAT.

Table 43: Default content of input data sets 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	Direct gateway output values 3	Not assigned
Byte 1	Input values, module 2	Input values, module 12	Output values, module 10	Direct gateway output values 4	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	Direct gateway output values 1	Not assigned	Not assigned
Byte 9	Input values, module 10	Output values, module 8	Direct gateway output values 2	Not assigned	Not assigned
Length	10 bytes	10 bytes	10 bytes	10 bytes	10 bytes

Detailed information on the content of the process image: see "Data transferred to the network (network input data sets)", page 128.

For information on how the process image is configured, see "Process image configuration", page 140 and the operating instructions titled “Flexi Soft in the Flexi Soft Designer Configuration Software” (SICK part number 8012998).

NOTE

The process data can also be read using the CoE objects 2000h and 2001h (see "CAN application layer over EtherCAT (CoE)", page 65). Simple access via SDO is recommended for diagnostic purposes. The faster PDO communication should be used in normal operation.

4.1.4.4 Output data – Network to Flexi Soft

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

NOTE

- Each output data set contains 10 bytes.
- The content of the output data sets can be configured in the configuration software.

4.1.4.5 Exporting tag names

Flexi Soft Designer makes it possible to export the tag names of the bits used in the input and output data sets. In the PLC, you can edit the start addresses of the data sets being used before exporting. The exported tag names and start addresses can then be imported as variables into the application program in the EtherCAT network configuration tool (e.g., TwinCAT PLC). This accelerates the programming of the PLC and makes it easier to identify individual bits in the EtherCAT PDOs.

Exporting tag names

- ► The tag names for the input and output data sets must be exported separately. To export the tag names for the input data sets, open the configuration page for **Flexi Soft to EtherCAT**. To export the tag names for the output data sets, open the configuration page for **EtherCAT to Flexi Soft**.
- ► Click on the **Export** button in the toolbar.
- ► Select the destination.
- ► Enter a name for the export file.
- ► In the bottom selection list, select the desired file format (e.g., *.csv or *.exp for TwinCAT).
- ► Click on **Save** to export the data.

NOTE

- The export function generates a 10-byte structure for each data set used with the byte or module names and a bit variable for each bit used.
- The name of the bit variable consists of the application name, the name of the byte and the name of the bit.
- We recommend assigning a tag name to each module in the Flexi Soft configuration and using a distinct tag name for all modules, bytes, and bits. Special characters in the tag names are deleted. Blank spaces are replaced with an underscore (“_”).
- The start addresses of the data sets in the TwinCAT PLC process image can be changed in the gateway configuration menu (see below).

Changing the start addresses of the data sets

- ► Start Flexi Soft Designer and load the hardware configuration, including the EtherCAT gateway. Then disconnect Flexi Soft Designer from the Flexi Soft system.
- ► Click on the **Interfaces** button above the main window and select the FX0-GETC or double-click on the FX0-GETC in the **Hardware configuration** to open the dialog box for gateway configuration.
- ► Click on the **Gateway configuration** button. The following dialog box is displayed:

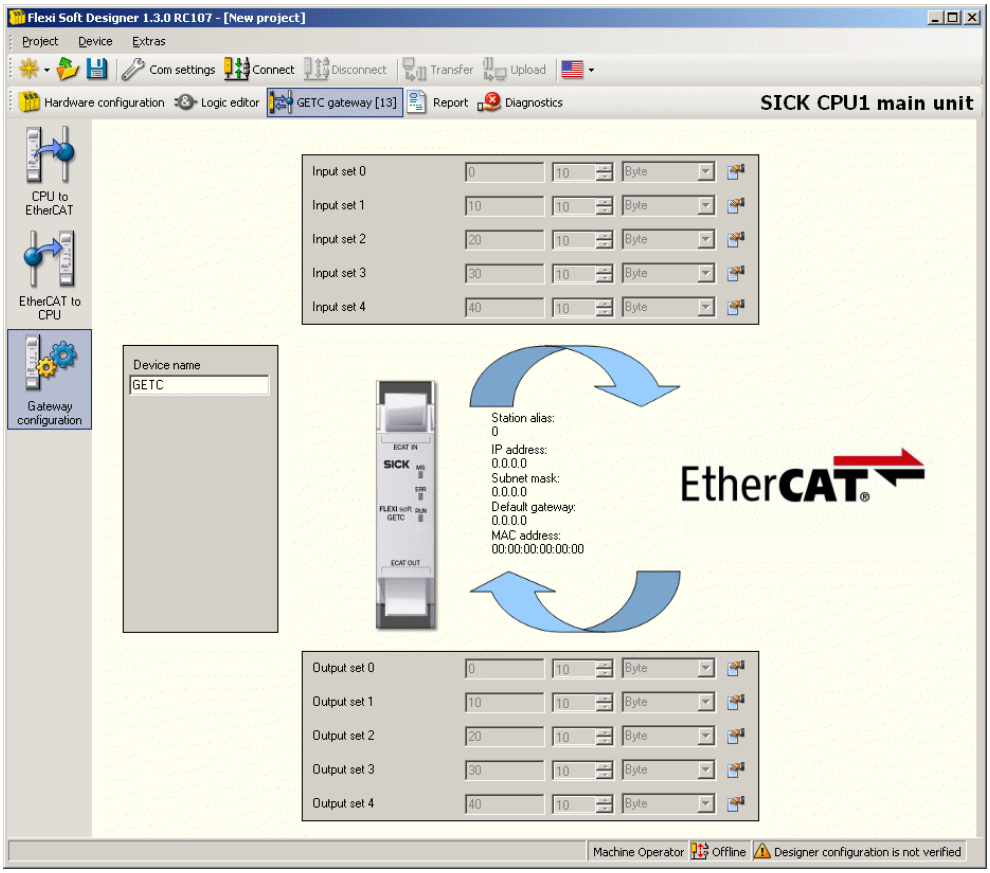

Figure 18: Configuration window for the EtherCAT gateway

► Click on the button to the right of the data set to be changed. The following dialog box is displayed:

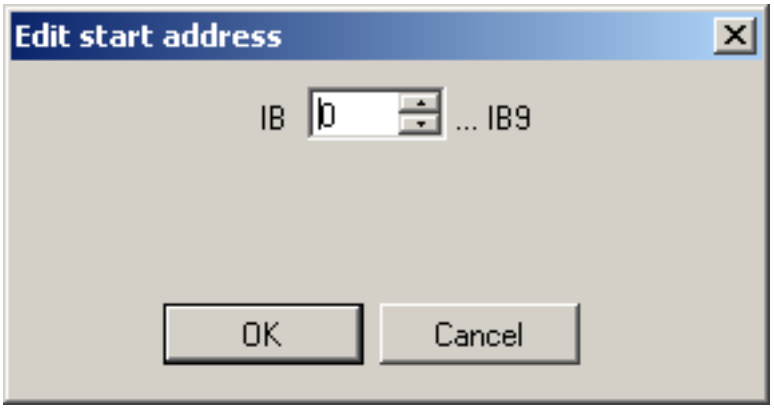

Figure 19: Changing data set start address

► Enter the new start address or change the start address using the arrows. The set start address is automatically checked for plausibility. It is not possible to configure data sets with overlapping address ranges.
► Click on **OK** to apply the new start address.

4.1.4.6 Activating Ethernet over EtherCAT (EoE)

Overview

The EoE function of the gateway needs to be activated using the EtherCAT network configuration tool (e.g., TwinCAT). The gateway itself does not have a real MAC address. For this reason, a virtual MAC address and the IP settings must be assigned to the device.

Prerequisites

- The gateway is in the pre-operational state or higher to enable the EoE protocol to access the EtherCAT mailboxes.

Procedure

Activating EoE

► Follow the instructions for activating EoE provided by the online help system or the user manual for the EtherCAT network configuration tool.
► Assign a virtual MAC address and the IP settings.

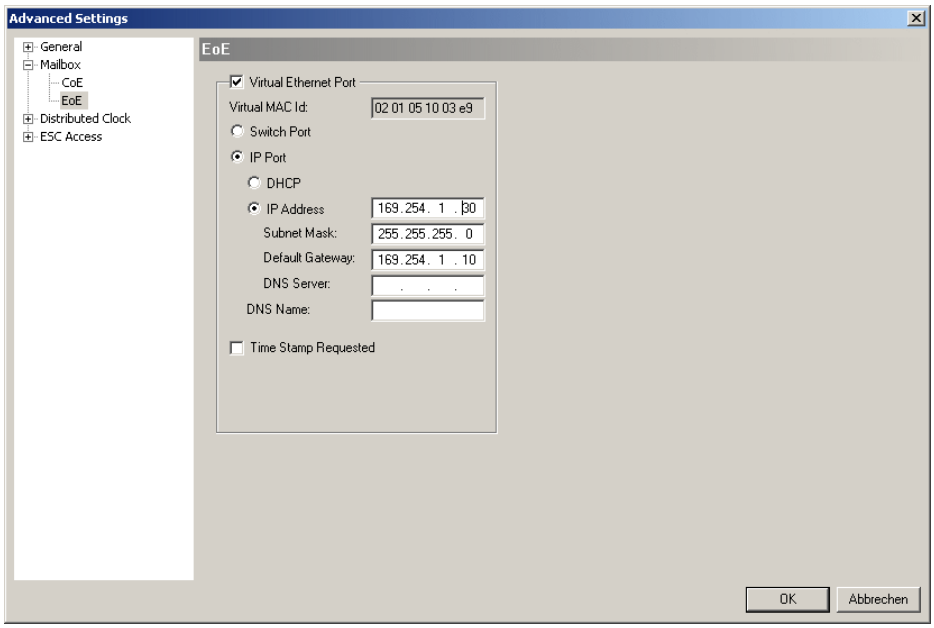

Figure 20: Activating EoE for the FX0-GETC in TwinCAT

► Loading the EoE configuration in the gateway.
✓ The gateway can be access via Ethernet.

Deactivating EoE

► Follow the instructions for deactivating EoE provided in the online help system or the user manual for the EtherCAT network configuration tool.
► Restart the Flexi Soft safety controller.
✓ EoE is deactivated.

4.1.4.7 TCP/IP configuration interface

The FX0-GETC must be part of a functioning EtherCAT network in order for the TCP/IP configuration network to be used. For this purpose, the EoE function of the FX0-GETC must be activated in the EtherCAT configuration tool and an IP address and a subnet mask must be assigned to it.

► Follow the instructions for assigning an IP address and a subnet mask provided by the online help system or the user manual for the EtherCAT network configuration tool.

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

► In the **Com settings** window, create a new TCP/IP profile or edit an existing TCP/IP profile for the FX0-GETC, check the **Enable TCP/IP routing** box, and enter a suitable **Subnet address** and **Gateway address**.

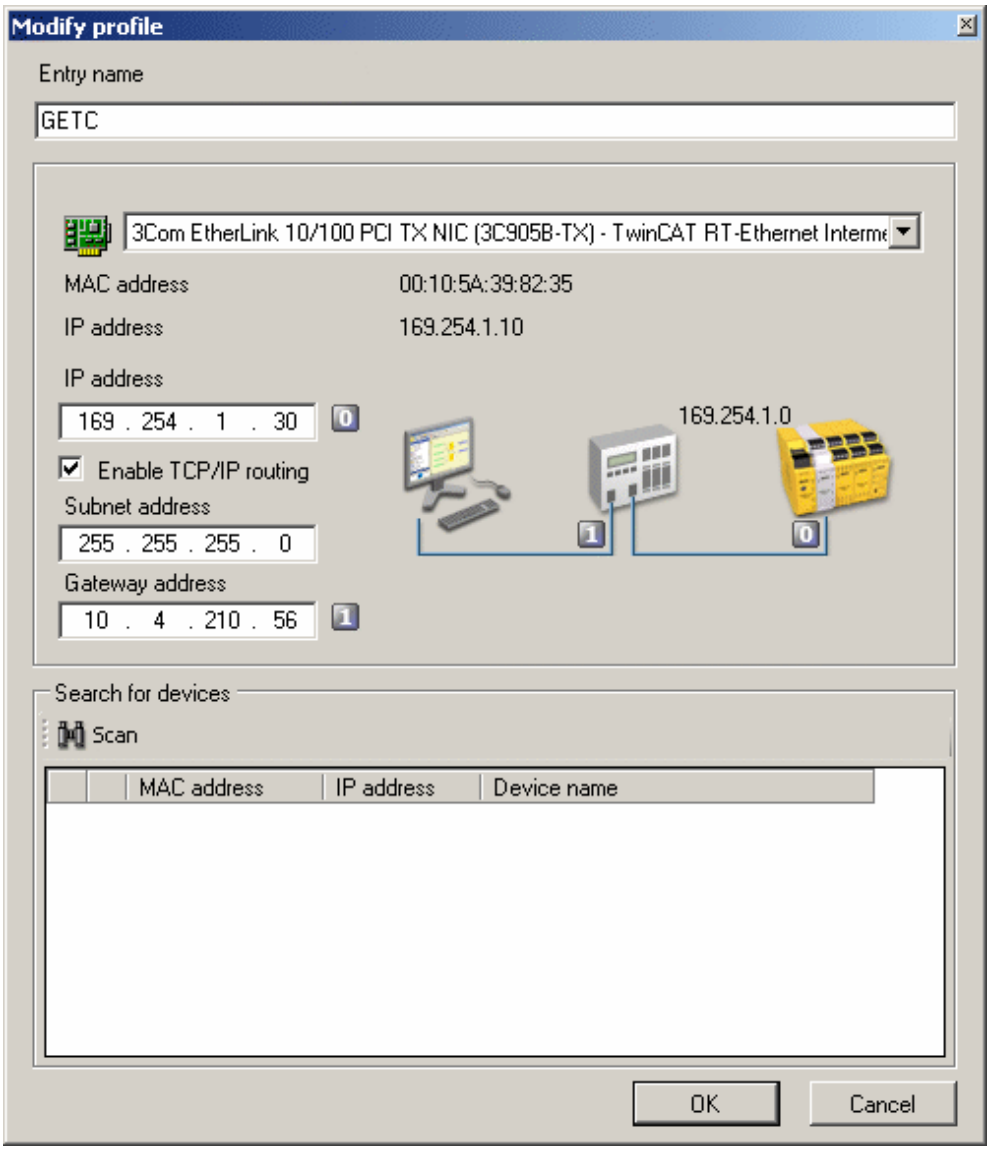

Figure 21: Activating TCP/IP routing for the FX0-GETC in the Flexi Soft Designer

NOTE

The **Default gateway** must be set correctly in the EoE settings of the EtherCAT gateway in the EtherCAT network configuration tool so that a connection can be established.

4.1.4.8 CAN application layer over EtherCAT (CoE)

CoE objects

The FX0-GETC supports several CoE objects. They can be displayed 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 also has a number of manufacturer-specific objects.

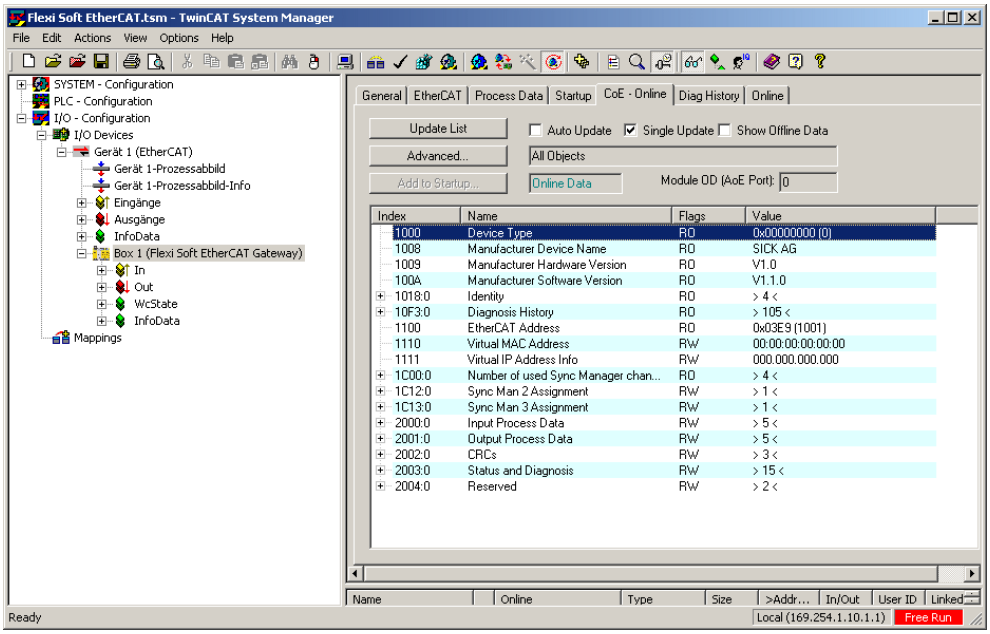

Figure 22: CoE object catalog of the FX0-GETC in TwinCAT

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 by means of SDO commands.

Input process data (2000h)

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

Table 44: Input process data of the FX0-GETC in 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 it available for acyclic use. It corresponds to the EtherCAT output PDO data of the FX0-GETC.

Table 45: Output process data of the FX0-GETC in 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

Checksums (2002h)

This object contains the checksums (description: see "Configuration checksums", page 133).

Table 46: Checksums of the FX0-GETC in CoE object 2002h

Index	Subindex	Name	Size
2002h	01	Overall checksum	4 bytes
	02	Flexi Soft checksum	4 bytes
	03	FX3-CPU0 and FX3-CPU1: reserved FX3-CPU2 and FX3-CPU3: ACR checksum	4 bytes

Status and diagnostics (2003h)

This object contains the module status bits of the Flexi Soft system. Each Flexi Soft module has 32 status bits, whereby each stands for a possible error message of the module. The significance of each bit depends on the module type (see "Error and status information of the modules", page 133).

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

Table 47: Status and diagnostics of FX0-GETC in CoE object 2003h

Index	Subindex	Name	Size
2003h	01h	Main module	4 bytes
	02h	Module 1	4 bytes
	03h ... 0Dh	Module 2 ... Module 12	4 bytes
	0Eh	Gateway 1	4 bytes
	0Fh	Gateway 2	4 bytes

Reserved (2004h)

This object is reserved for future use.

Diagnostic History (10F3h)

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

Subindex 1 of the **Diagnostic History** contains the maximum number of possible diagnostic history entries. Subindex 2 (Newest) references the newest diagnostic message. Subindex 3 (Acknowledged) references the last message that was acknowledged or – if no messages have been acknowledged yet – the last entry. Subindex 4 is True if Reading is necessary (i.e. if Newest and Acknowledged are different).

Table 48: Structure of the Diagnostic History object

Subindex	Content	Format	Comments
01h	Max. entry number	USINT	–
02h	Newest	USINT	Subindex of the newest history entry (e.g. 2Ah)
03h	Acknowledged	USINT	Subindex of the last acknowledged history entry
04h	Reading required	BOOLEAN	True, if Newest is not the same as Acknowledged
05h	Flags	UINT	Flags to control the transmission and saving of diagnostic messages – the Flexi Soft EtherCAT gateway does not support any of the optional options.
06h ... 69h	Diagnostic history entries	OCTET STRING	See below

Each diagnostic message consists of a diagnostic code and ASCII character string containing the parameter set of the message.

- The diagnostic code consists of the module number and the diagnostic bit of the respective module.
- The ASCII character 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 diagnostic message. Incoming diagnostic messages are identified with a "+" and outgoing messages with a "–".

If a problem has been diagnosed and then solved, the object 10F3h contains two diagnostic messages which differ only by the trailing "+" or "–".

The **Diagnostic History** object is structured as a ring buffer. If subindex 69h has been written, the next entry starts again with subindex 06h.

If the number of diagnostic messages that have not been acknowledged reaches 100, older messages will not be overwritten. Instead, the newest diagnostic message will be replaced by a buffer overflow error message (FFFFh).

The FX0-GETC does not support a time stamp for the **Diagnostic history** object. If a time stamp is required, then the reading device (e.g., the PLC) can be programmed in such a way that it adds the time stamp when reading a diagnostic message.

Detailed information on the structure and use of this object is contained in the document "EtherCAT Protocol Enhancements" of the EtherCAT Technology Group (ETG.1020), which is available online at www.ethercat.org.

4.1.5 The TCP/IP configuration interface

The Flexi Soft Ethernet gateways feature a TCP/IP configuration interface for configuring the Flexi Soft system via Ethernet TCP/IP. The interface works in parallel with Ethernet TCP/IP or other Ethernet protocols.

WARNING
Configuration, diagnostics or operation errors due to several simultaneous configuration connections

The dangerous state may not be stopped or not be stopped in a timely manner in the event of non-compliance.

► Dot not establish concurrent configuration connections to a Flexi Soft system. This applies regardless of the configuration software used and the selected interface (RS-232, Ethernet, USB).

NOTE

Remote TCP/IP connections with excessive signal propagation times can be unstable. Signal propagation times > 300 ms may result in the connection being terminated.

- ► Take account of signal propagation times in the case of remote TCP/IP connections.
- ► Use the ping command to check the signal propagation time to the gateway.
- ► Make sure that the connection is fast enough or change the routing (if possible).

Or:

- ► Use a piece of remote maintenance software such as TeamViewer to control the local computer that has the configuration software installed on it and is connected locally to the Flexi Soft system.

Or:

- ► Contact SICK Support.

Carry out the following steps to configure a gateway for TCP/IP for the first time:

Step 1: Assign an IP address

- ► Connect a computer to the RS-232 interface of the main module.
- ► Switch on the Flexi Soft system.
- ► Launch the Flexi Soft Designer configuration software that has been installed on the computer and load the hardware configuration, including the gateway.
- ► Click on the **Disconnect** button to switch to offline mode.
- ► Click on the **Interfaces** button above the main window and select the required gateway or double-click on the required gateway in the **Hardware configuration** to open the dialog box for gateway configuration.
- ► Click on the **Gateway configuration** button.

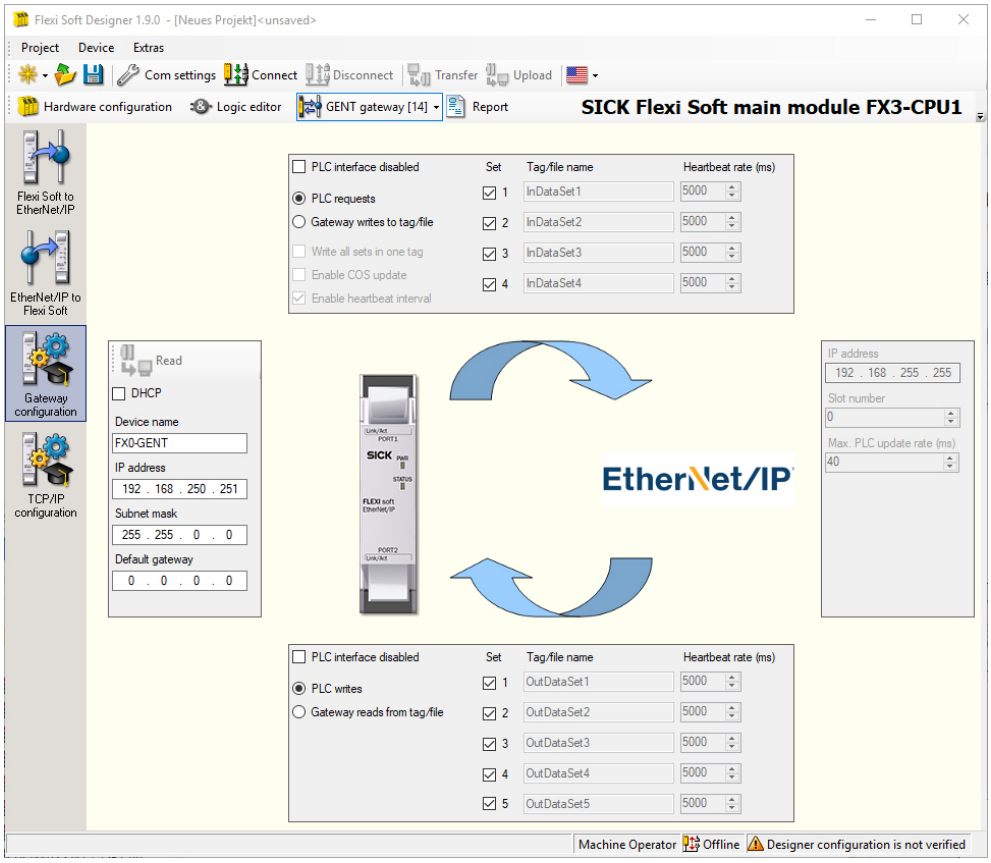

Figure 23: Configuration window for the Ethernet gateway

On the left-hand side of the dialog box, you will find the area for the gateway IP configuration.

- ► If you wish, enter a **Device name** for the Flexi Soft gateway.
- ► Enter a valid **IP address** for the Flexi Soft gateway and, if necessary, a valid **Subnet mask**, and a valid IP address for a **Default gateway**.

Or:

- ► If the network uses a DHCP server, check the **DHCP** box.

NOTE

The FX0-GPNT with firmware version ≥ V3.00.0 does not support DHCP.

- ► Click on **Connect** to switch to online mode.
- ► Click on **Transfer** to transfer the configuration to the Flexi Soft system.

NOTE

- If the project is online, the current gateway IP settings can be read out. To do this, click on the **Read** button in the top left corner of the area for the gateway IP configuration.
- The preset IP address of the gateway on delivery is 192.168.250.250. The preset IP address is also printed on the gateway type label.
- For gateway types FX0-GENT and FX0-GPNT with firmware version ≥ V3.00.0, the preset IP address on delivery is 0.0.0.0.

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

- ► Connect one of the two gateway Ethernet connections to the Ethernet network using a shielded Ethernet cable.
- ► Connect a computer to the same Ethernet network. Make sure that the computer IP address settings match the network configuration.

NOTE

The computer can also be directly connected to one of the two gateway Ethernet connections. In this case, either the computer IP address settings or those of the gateway must be adjusted to match the settings of the other device.

- ► Launch the Flexi Soft Designer configuration software that has been installed on the computer and load the hardware configuration, including the gateway.
- ► Click on the **Disconnect** button to switch to offline mode.
- ► Click on **Com settings**. The following dialog box is displayed:

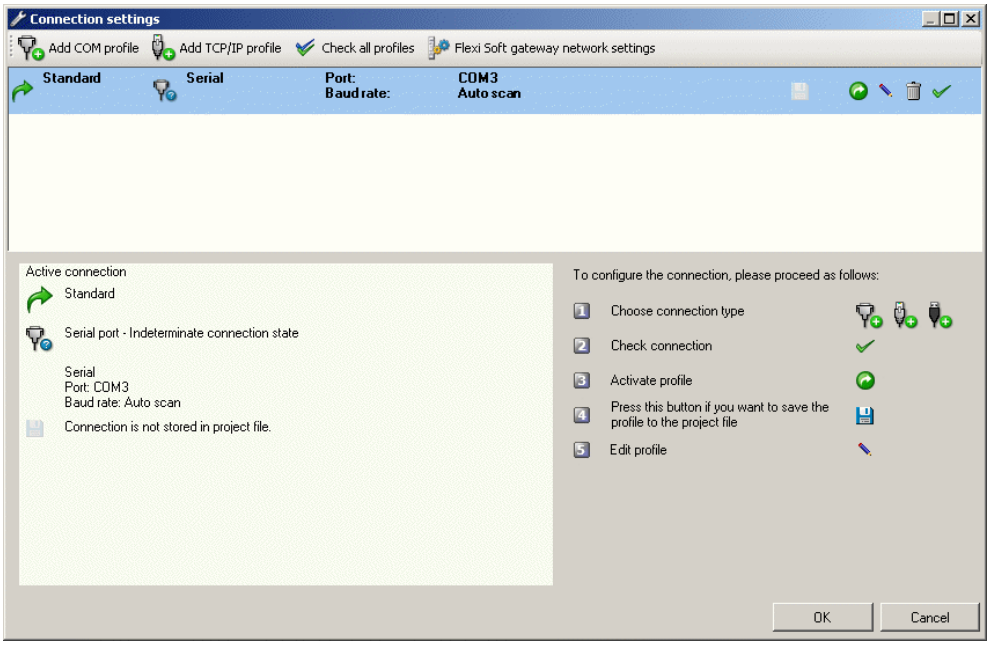

Figure 24: Communication settings dialog box

- ► Click on **Add TCP/IP connection profile**. The following dialog box is displayed:

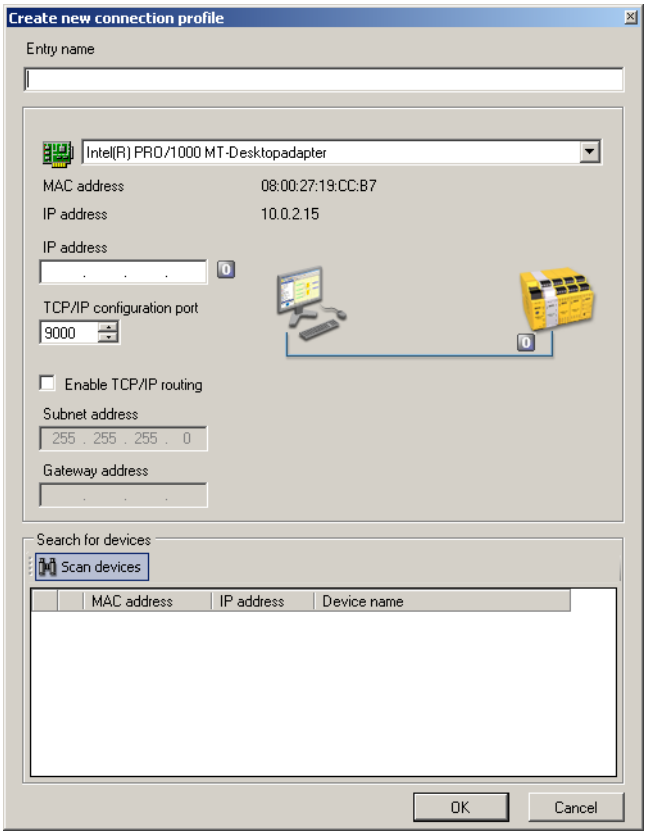

Figure 25: Dialog box for adding a new TCP/IP profile

► Click on **Search for devices** to search for the Flexi Soft gateways in the Ethernet network. The IP addresses, the MAC addresses, and the device names of all found gateways are displayed in the dialog box.

NOTE

- Flexi Soft Designer version ≥ V1.4.0 performs a 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 are identified, even if they are located in another subnetwork.
- A FX0-GETC can also be detected if was previously configured for EoE (see "Activating Ethernet over EtherCAT (EoE)", page 64 and see "TCP/IP configuration interface", page 64).
- The Flexi Soft Designer with a version < V1.4.0 can detect only gateways with a matching subnet address.

► Select the gateway that is to be used for the access.
► Enter a name for the access in the **Access name** input field.
► Set the desired number for the **TCP/IP configuration port** if required. Port 9000 is preset.

NOTE

- The TCP/IP configuration port in the device always stays as port 9000, regardless of the configuration in Flexi Soft Designer. Ethernet switches with IP masquerading or PAT (port and address translation) can be supported by configuring the port number in Flexi Soft Designer.
- The Flexi Soft Designer with version ≥ 1.8.0 is required in order to change the number for the **TCP/IP configuration port**.

► Click on **OK**. The new access is displayed in the **Communication settings** dialog box:

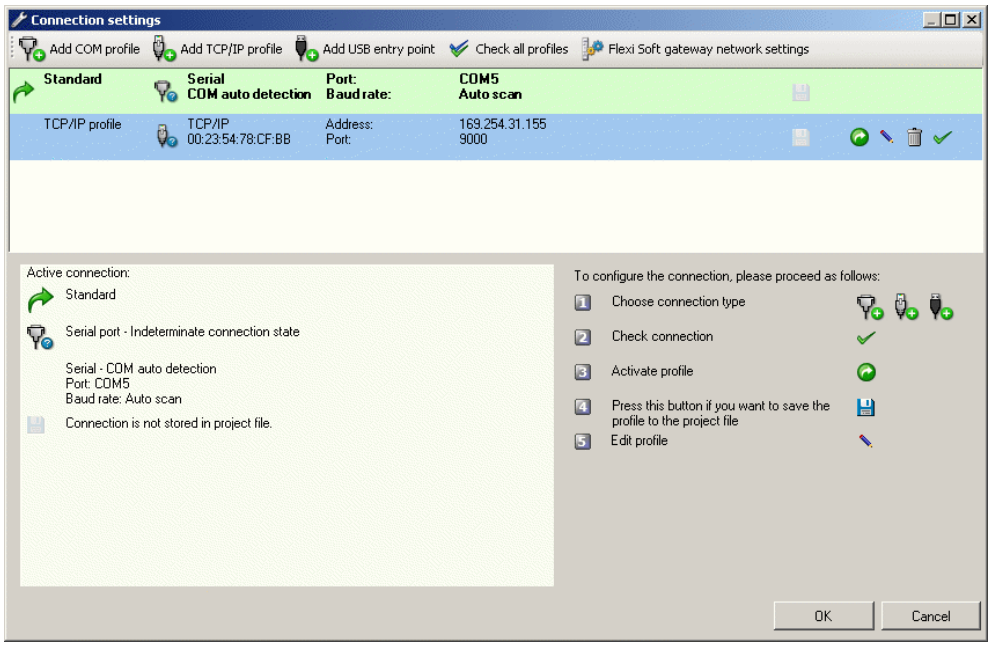

Figure 26: Communication settings dialog box with new TCP/IP profile

This access must be activated in order to use it.

► Click on the **Activate profile** button (white arrow in green circle) on the right side. The access is now activated and marked accordingly.

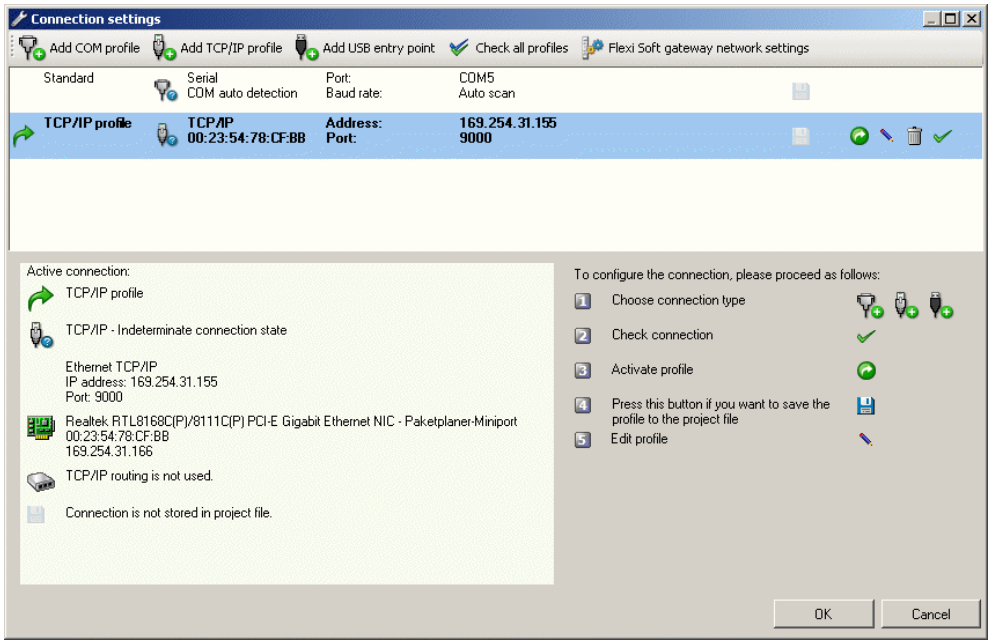

Figure 27: Communication settings dialog box with activated new TCP/IP profile

► Click on **OK**. All communication with the Flexi Soft system will now take place via TCP/IP. To use the access via the serial interface again, this will first need to be reactivated.

Step 3: Establishing a connection via TCP/IP

► Click on the **Connect** button to switch to online mode.

Changing the network settings for a Flexi Soft gateway

► In the **Communication settings** dialog box, click on the **Flexi Soft gateway network settings** button. The **Search for devices** dialog box opens.

► Click on the **Search for devices** button. The software scans the network for connected gateways and displays any devices that it finds in the list.

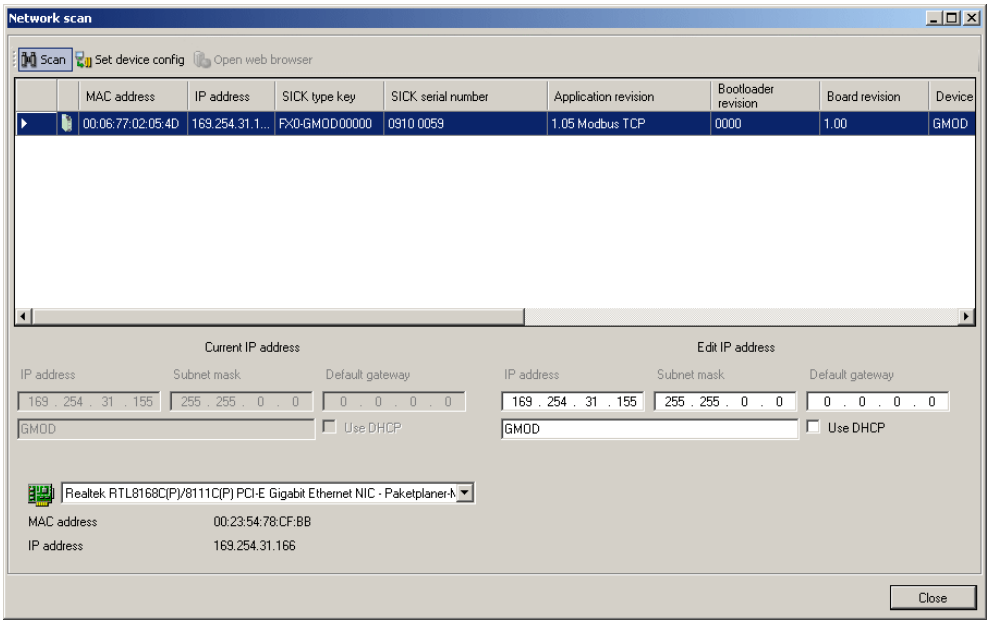

Figure 28: Gateways found in the Search for devices dialog box

NOTE

- Flexi Soft Designer version ≥ V1.4.0 performs a 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 are identified, even if they are located in another subnetwork.
- A FX0-GETC can also be detected if was previously configured for EoE (see "Activating Ethernet over EtherCAT (EoE)", page 64 and see "TCP/IP configuration interface", page 64).
- The Flexi Soft Designer with a version < V1.4.0 can detect only gateways with a matching subnet address.

► Click on the required gateway.

► 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 finds a gateway of the Flexi Classic product family in the network, this will also be shown in the list. These gateways are equipped with an internal web server and can be addressed using the **Open web browser** button.

4.1.6 Ethernet TCP/IP socket interface

Ethernet TCP/IP socket interface

The FX0-GENT, FX0-GMOD, and FX0-GPNT Ethernet gateways each support four TCP/IP socket interfaces. This enables multiple applications to communicate with the gateway via Ethernet TCP/IP at the same time.

The number of possible connections depends on the firmware version of the gateway.

Table 49: Number of possible TCP/IP connections

Firmware version	Number of connections possible per socket	Total number of possible connections
V1.xx.x	1 per socket	4
≥ V2.00.0	6 per socket	24

The specific network interface for the relevant gateway (e.g., Modbus® TCP) functions in parallel. As a result, the TCP/IP socket configuration is not affected by the configuration or operation of this interface and is executed regardless.

4.1.6.1 Configuring the Ethernet TCP/IP socket interface

Overview

The gateway processes the data of a Flexi Soft system and makes it available in different data sets via the TCP/IP socket interface.

Important information

NOTE

Use different output data set numbers for different PLC connections or TCP/IP sockets.

The output data set of the Ethernet gateway can be changed simultaneously via several communication interfaces or TCP/IP sockets (e.g. Modbus TCP and Ethernet TCP/IP) if the same output data set number is used for the different connections. In this case, the last message overwrites the data received earlier. To prevent this, a output data set number must be used for each connection.

Procedure

Configuring the Ethernet TCP/IP socket interface

1. Start the configuration software and load the hardware configuration, including the gateway.
2. Click on the **Interfaces** button above the main window and select the required gateway or double-click on the required gateway in the **Hardware configuration** to open the dialog box for gateway configuration.

3. Click on **TCP/IP configuration**. The following dialog box is displayed:

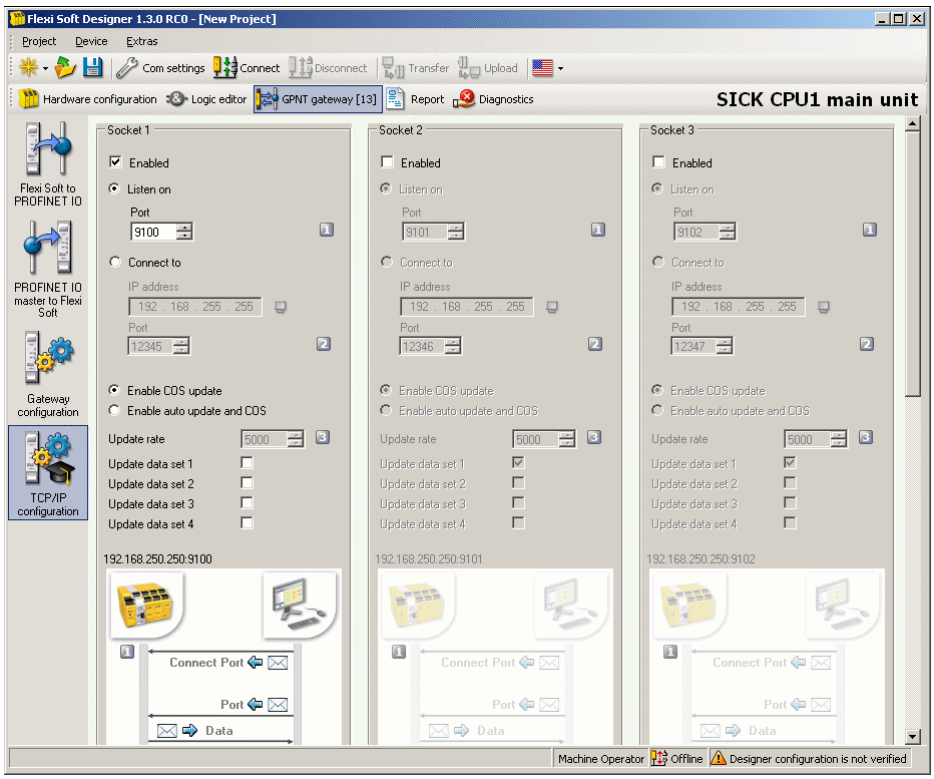

Figure 29: TCP/IP configuration dialog box

If the gateway needs to establish connections with external applications, then it must be configured as follows:

1. Select the **Connect to** option.
2. Under **IP address**, enter the IP address of the computer on which the application is running.
3. Enter the port of the application.

   NOTE | The configuration is incorrect if the IP address and/or the port is set to zero in **Connect to** mode.

If external applications need to establish connections with the gateway, then the gateway must be configured as follows:

1. Select the **Listen on** option.
2. Enter the port of the application

   NOTE |

   - 9100 to 9103 are the recommended port numbers (default).
   - Port 0 and Port 9000 are reserved and must not be used (incorrect configuration).
   - Port numbers 0 to 1023 are managed by the Internet Assigned Numbers Authority (IANA) and should not be used in order to avoid collisions (see www.iana.org/assignments/port-numbers).
   - Port 65535 is not supported by the FX0-GENT 3.xx, FX0-GPNT 3.xx and FX3-GMOD 3.xx gateways.
3. Select how the data are to be transmitted.

Further topics

- "Data transferred to the network (network input data sets)", page 128

4.1.6.2 Transmission type

Polling mode and auto update mode

When a TCP/IP socket connection has been established (either by an application on one computer or by the gateway itself), there are two possible ways in which the data sets can be transferred:

- Polling mode: The application requests data sets with a control command.
- Auto update mode: The gateway updates the data sets according to the configuration (gateway writes to address/port).

In auto update mode, there are two possible update modes for the gateway to update the data:

- COS update (Change of State): when any data of the input data set changes its status
- Auto update: Data is sent according to the configured **Update rate** in milliseconds.

NOTE

A change of state (COS) triggers an immediate data update – irrespective of the configured **Update rate**, i.e., COS is always active.

General structure of the messages

The request/response (e.g. telegram) has the following structure:

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

Table 50: Message structure

Parameter	Length	Description
Command	WORD	0h = Undefined (no command) **Polling-specific** 00F1h = Input data set request 001Fh = Input data set response **Auto update-specific** 00E1h = Auto update control 001Eh = Response to auto update control 002Eh = Auto update message of the input data set(s) **Reading/writing to the digital outputs** 00F2h = Write settings output data set 002Fh = Response to write settings output data set
Parameter	Length depends on command	As defined in the respective command
Data	Length depends on command	As defined in the respective command

Error response for invalid messages

The gateway sets the most significant bit of the command word if an invalid or incorrectly formatted message is received.

Table 51: Error response

Parameter	Length	Description
Command	WORD	Bit 15 of the received command is set (i.e. the command 00F2h becomes 80F2h).

Parameter	Length	Description
Following data	Length depends on command	Unchanged. Returned as received

4.1.6.3 Configuring polling mode

Overview

In this mode, the gateway only sends data on request (polling). The application must send messages to request a data set. The gateway then responds with messages.

Important information

NOTE
To prevent the connection from being terminated automatically, the application must request data at least every 30 s in polling mode.

Procedure

1. Start the configuration software and load the hardware configuration, including the gateway.
2. Click on the **Interfaces** button above the main window and select the required gateway or double-click on the required gateway in the **Hardware configuration** to open the dialog box for gateway configuration.
3. Click on **TCP/IP configuration**. The following dialog box is displayed:

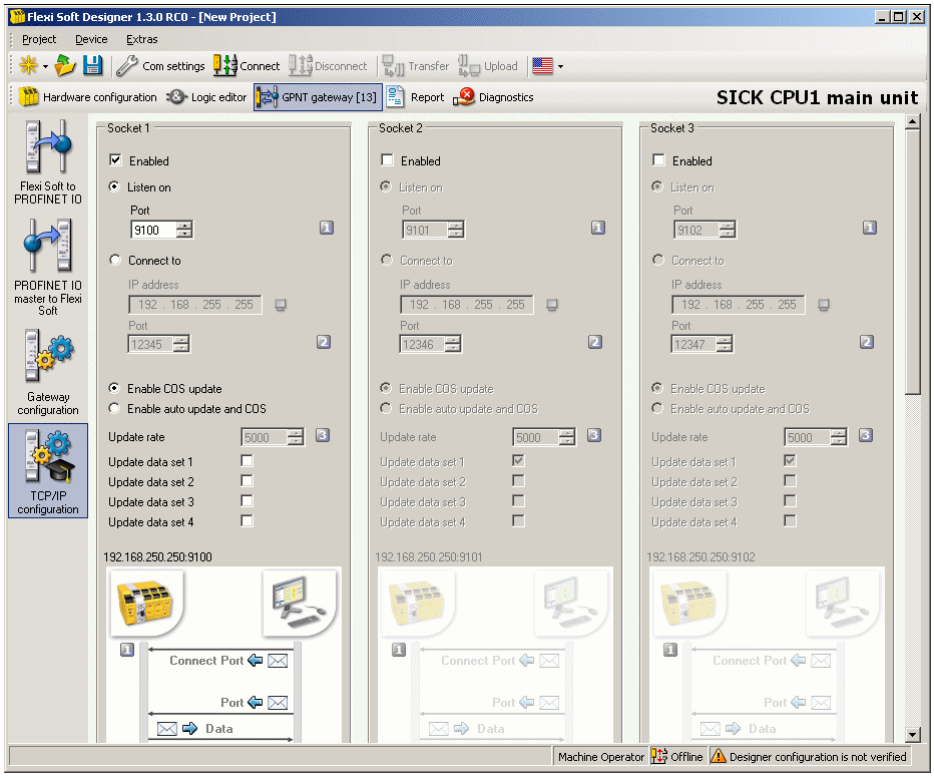

Figure 30: TCP/IP configuration for polling mode (application requests)

4. Select the **Listen on** option.
5. Enter the port to which the application is connected.
6. For polling mode, select the **Enable COS update** mode.
7. Deselect all **Update data set n** checkboxes.

4.1.6.3.1 Input data sets

Request for input data set

The request is sent to the gateway by an application. The message for requesting an input data set must have the following structure:

Table 52: Request for input data set

Parameter	Length	Value
Command	WORD	00F1h = Data set request
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

Response to data set request

The gateway sends the application a response that is structured as follows:

Table 53: Response to data set request

Parameter	Length	Value
Command	WORD	001Fh = Data set response
Length of data set 1	WORD	0 = Data set is not send back in the data set data field. Non-zero = Length of the data set
Length of data set 2	WORD	0 = Data set is not send back in the data set data field. Non-zero = Length of the data set
Length of data set 3	WORD	0 = Data set is not send back in the data set data field. Non-zero = Length of the data set
Length of data set 4	WORD	0 = Data set is not send back in the data set data field. Non-zero = Length of the data set
Data set data	Byte array	Data set information

4.1.6.3.2 Output data sets

Command for writing the output data sets

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

Table 54: Command for writing the output data sets

Parameter	Length	Value
Command	WORD	00F2h = Command for writing the output data sets
Output data set 1 length	WORD	0 = Output data set is not contained in data set data field. Not zero = Length of data set
Output data set 2 length	WORD	0 = Output data set is not contained in data set data field. Not zero = Length of data set

Parameter	Length	Value
Output data set 3 length	WORD	0 = Output data set is not contained in data set data field. Not zero = Length of data set
Output data set 4 length	WORD	0 = Output data set is not contained in data set data field. Not zero = Length of data set
Output data set 5 length	WORD	0 = Output data set is not contained in data set data field. Not zero = Length of data set
Data set data	Byte array	Data set information

Response to writing the output data sets

The gateway sends the application a response that is structured as follows:

Table 55: Response to writing the output data sets

Parameter	Length	Value
Command	WORD	002Fh = Response to the message to write the settings of the output data sets
Status	WORD	0 = Success – output data sets were written correctly. 1 = Error – output data sets can not be written for one of the following reasons: • Interruption in backplane communication • Incorrect routing information

4.1.6.4 Commands for configuring auto update mode

Overview

The gateway can be configured so that it automatically updates the data set information (i.e. the application does not need to send requests like in polling mode) as soon as the connection to the application has been established.

The configuration settings are available via the configuration software or the TCP/IP socket interface itself. Using one interface will not deactivate the other. Auto update mode could, for example, be activated via the configuration software and deactivated with a TCP/IP command.

Activation of auto update mode

An application sends the following command to the gateway in order to configure auto update mode. The command can be used to activate or deactivate auto update mode directly via the TCP/IP socket interface.

Table 56: Command for configuring auto update mode

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

Parameter	Length	Value
Update frequency in heartbeat mode	WORD	0 = Deactivate heartbeat messages Not zero = Activate heartbeat message at a specific frequency in milliseconds Minimum = 40 ms

Deactivation of auto update mode

Auto update mode is deactivated if all input data set request flags are set to zero.

Response to configuring auto update mode

The following response is sent by the gateway to the application:

Table 57: Response to configuring auto update mode

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

4.1.6.5 Configuring auto update mode

Procedure

1. Start the configuration software and load the hardware configuration, including the gateway.
2. Click on the **Interfaces** button above the main window and select the required gateway or double-click on the required gateway in the **Hardware configuration** to open the dialog box for gateway configuration.
3. Click on **TCP/IP configuration**. The following dialog box is displayed:

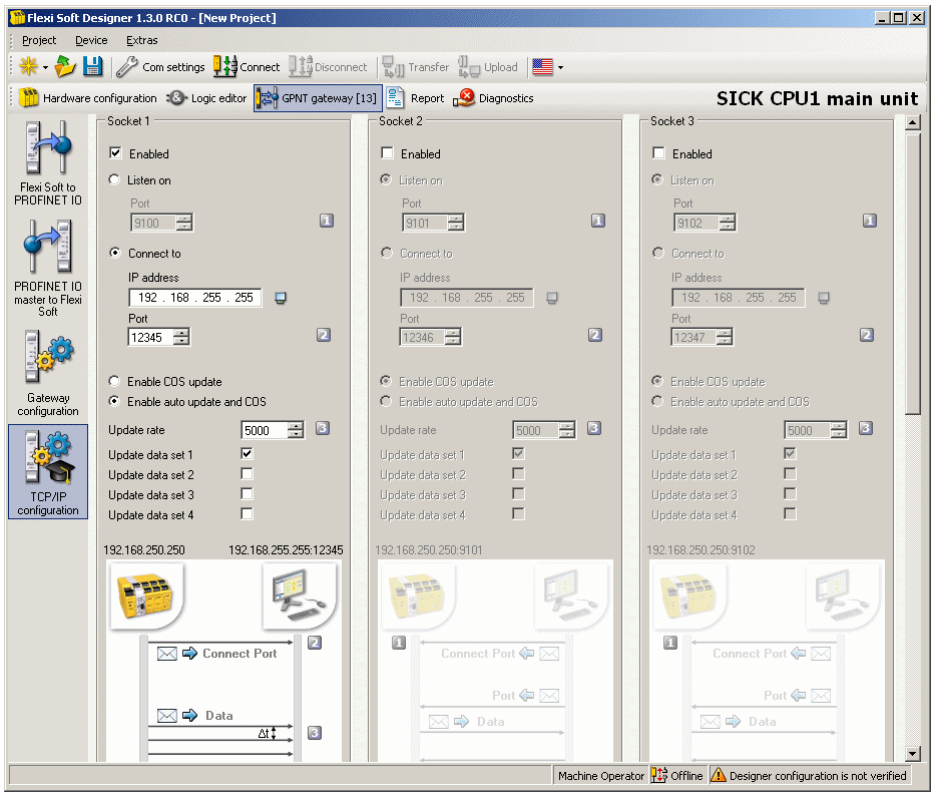

Figure 31: TCP/IP configuration for auto update

4. Select the **Connect to** option.
5. Enter the IP address and the port to which the gateway is to write.
6. Select update mode.

7. If you wish to use auto update and COS, enter the update rate in milliseconds.
8. Select which data is to be updated: Select the **Update data set n** checkbox.

4.1.6.5.1 Auto update mode message

The following message is sent by the gateway to the application when it is operated in auto update mode.

Table 58: Message in normal operation of auto update mode

Parameter	Length	Value
Command	WORD	002Eh = Auto update message of the data set(s)
Data set 1 length	WORD	0 = Data set is not returned in data set data field. >0 = Length of data set
Data set 2 length	WORD	0 = Data set is not returned in data set data field. >0 = Length of data set
Data set 3 length	WORD	0 = Data set is not returned in data set data field. >0 = Length of data set
Data set 4 length	WORD	0 = Data set is not returned in data set data field. >0 = Length of data set
Data set data	Byte array (length dependent on the data set configuration)	Data set information

4.1.7 Example of a TCP/IP process image

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

Table 59: Example of a TCP/IP process image

Byte values [Hex]	Part of the message	Significance
00 2E	Command	Auto update of the data sets
00 32	Command parameter	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

Byte values [Hex]	Part of the message	Significance
03 FF 03 03	Data set 1 (default for byte assignments)	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 of module 3 ... 12
00 00 00 00 00 00 00 00 00 00 00 00		Output values of 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
52 A1 10 4C	Data set 2	Overall checksum (same value as the Flexi Soft checksum)
52 A1 10 4C		Flexi Soft checksum
00 00 00 00		FX3-CPU0 and FX3-CPU1: reserved FX3-CPU2 and FX3-CPU3: ACR checksum
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 [1]	Status, module 0 (main module): 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: Error in external test signal at input 3. Byte 3: FD = 11111101: Error: output 1 stuck at low
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF		Status of 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 of modules 7 ... 12 (no modules available)
FF FF FF FF		Status, module 13 (e.g. FX0-GENT): OK
FF FF FF FF		Status, module 14 (no module present)

Byte values [Hex]	Part of the message	Significance
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 00 00 00 00	Data set 4	Reserved

1) The data from data set 3 is transferred in little endian format, i.e., as a 32-bit double word with the least significant byte in the leftmost position.
Exception: If TCP/IP sockets are used with the FX0-GENT, data from data set 3 is transferred in big endian format, i.e., as a 32-bit double word with the most significant byte in the leftmost position.

4.2 Fieldbus gateways

This chapter describes how to configure the following gateways:

- FX0-GPRO PROFIBUS DP gateway
- FX0-GCAN CANopen gateway
- FX0-GDEV DeviceNet gateway

4.2.1 FX0-GPRO PROFIBUS DP gateway

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

4.2.1.1 Basic configuration – Setting the PROFIBUS address

The PROFIBUS DP address can be set either using the switches on the gateway or by using the configuration software.

Setting the PROFIBUS DP address using the hardware address switches

► Set the PROFIBUS DP address with the hardware address switches on the front of the device: Then switch off the Flexi Soft system and switch back on again after at least 3 seconds. Also refer to the operating instructions "Flexi Soft Gateways Hardware" (SICK part number 8012662).

Setting the PROFIBUS DP address using the Flexi Soft Designer configuration software

► Set the address switch on the front of the device to “00”.
► Start Flexi Soft Designer and load the hardware configuration, including the PROFIBUS DP gateway. Then disconnect Flexi Soft Designer from the Flexi Soft system.
► Click on the **Interfaces** button above the main window and select the FX0-GPRO or double-click on the FX0-GPRO in the **Hardware configuration** to open the dialog box for gateway configuration.
► Click on the **Gateway configuration** button. The following dialog box is displayed:

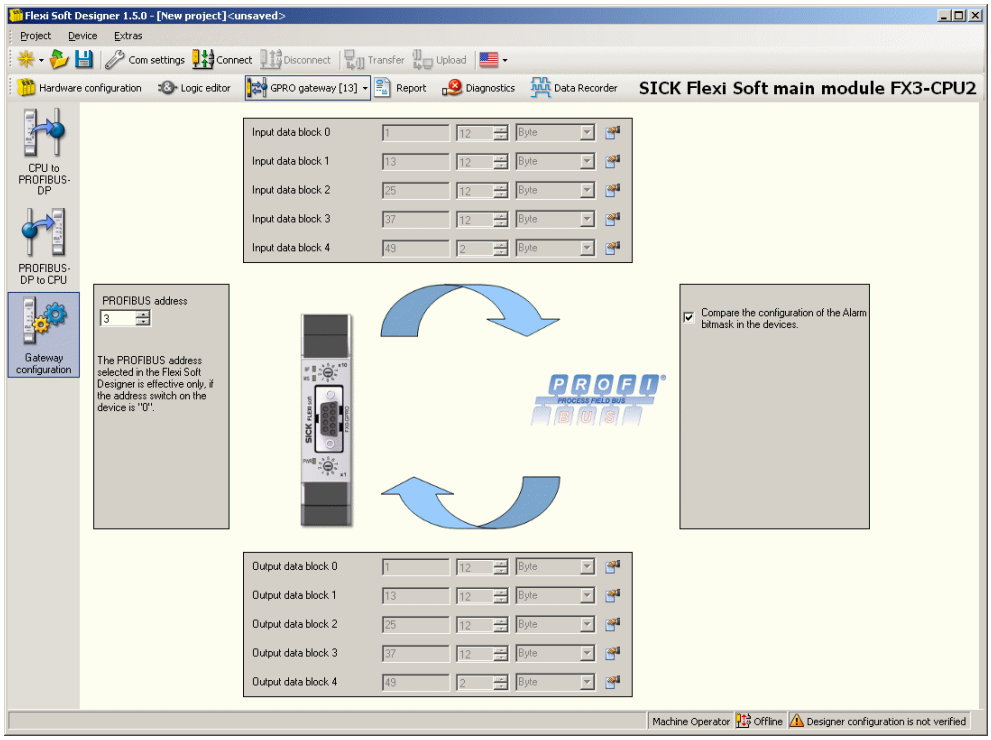

Figure 32: Setting the PROFIBUS DP address of the FX0-GPRO

- ► Enter the PROFIBUS address in the **PROFIBUS address** field.
- ► Click on **Connect** to switch to online mode.
- ► Click on **Transfer** to transfer the configuration to the Flexi Soft system.

NOTE

- A PROFIBUS address in the range 1 to 99 can be set by means of the hardware address switches.
- A PROFIBUS address in the range 3 to 125 can be set by means of the configuration software.
- The PROFIBUS master is not able to overwrite the PROFIBUS address.
- When you change an address setting, it only takes effect once the Flexi Soft system has been switched off and then back on again.
- The address set on the PROFIBUS DP gateway can be read out in online mode. To do this, click on the **Read** button above the **PROFIBUS address** field in the configuration window.

Data transmission rate

The data transmission rate is set automatically.

The maximum data transmission rate is 12 Mbit/s.

Configuration comparison of the alarm bitmask

This check box in the dialog box for the **Gateway configuration** is available for compatibility reasons and should not generally be changed. When loading configurations created with a version of Flexi Soft Designer < V1.3.1, the box is not checked. For configurations created with a version of Flexi Soft Designer > V1.3.1, the box is checked.

If this box is checked, the alarm bit masks in the device are updated by Flexi Soft Designer. This function is available for a FX0-GPRO with firmware version > V1.30.0. This makes it possible to adjust the scope of the alarms generated on the fieldbus side to new software versions of other modules.

If the status of this box is changed in a configuration that has already been verified, then this configuration must be verified again.

4.2.1.2 PROFIBUS – Data selection

Operating data transmitted by the FX0-GPRO

The GSD file of the FX0-GPRO provides input and output data blocks (virtual I/O device modules) that contain operating data. These five blocks have to be configured in a DP configurator in their natural numerical order (1, 2, 3, 4, 5). No other order is possible.

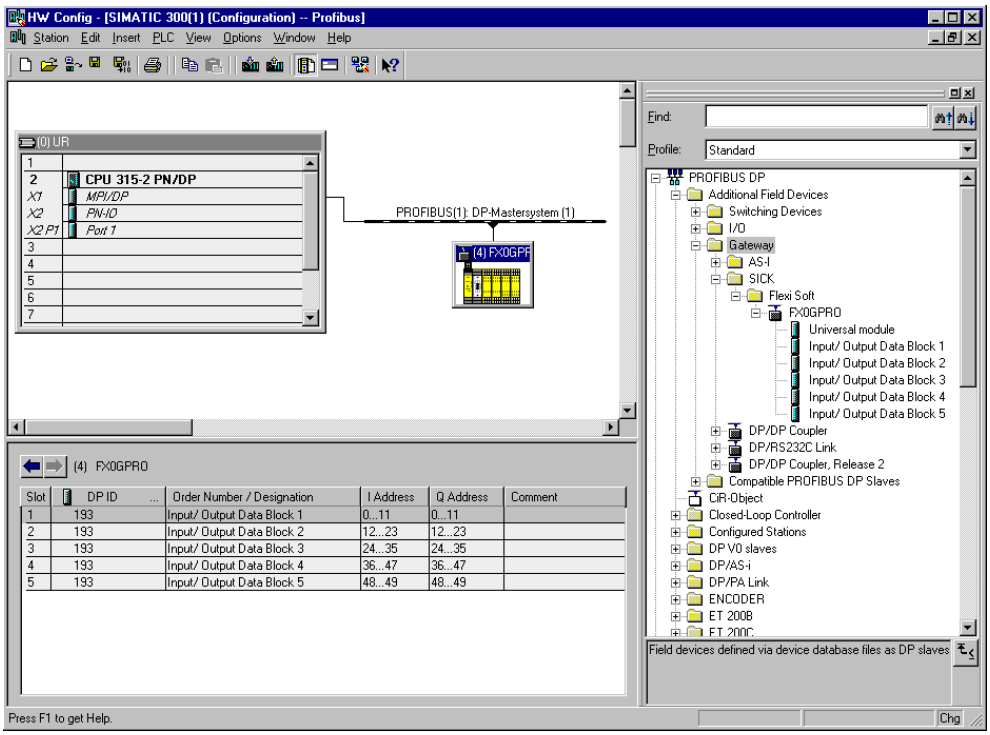

Figure 33: Example of a PROFIBUS DP configuration in the SIEMENS SIMATIC Manager software

NOTE

- Depending on what PLC you are using, further modules may be displayed (e.g. “universal module”). These modules are not required and should be ignored.
- Data blocks 1 to 4 each contain 12 bytes while data block 5 contains 2 bytes.
- The content of the data blocks can be freely selected but it is pre-configured.

Table 60: Default content of input data blocks 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

	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 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	Direct gateway output values 1	Not assigned	
Byte 5	Input values, module 6	Output values, module 6	Direct gateway output values 2	Not assigned	
Byte 6	Input values, module 7	Output values, module 7	Direct gateway output values 3	Not assigned	
Byte 7	Input values, module 8	Output values, module 8	Direct gateway output values 4	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

Detailed information on the content of the process image: see "Data transferred to the network (network input data sets)", page 128.

Setting the start address for the data blocks

- ► Start Flexi Soft Designer and load the hardware configuration, including the PROFIBUS DP gateway. Then disconnect Flexi Soft Designer from the Flexi Soft system.
- ► Click on the **Interfaces** button above the main window and select the FX0-GPRO or double-click on the FX0-GPRO in the **Hardware configuration** to open the dialog box for gateway configuration.
- ► Click on the **Gateway configuration** button. The following dialog box is displayed:

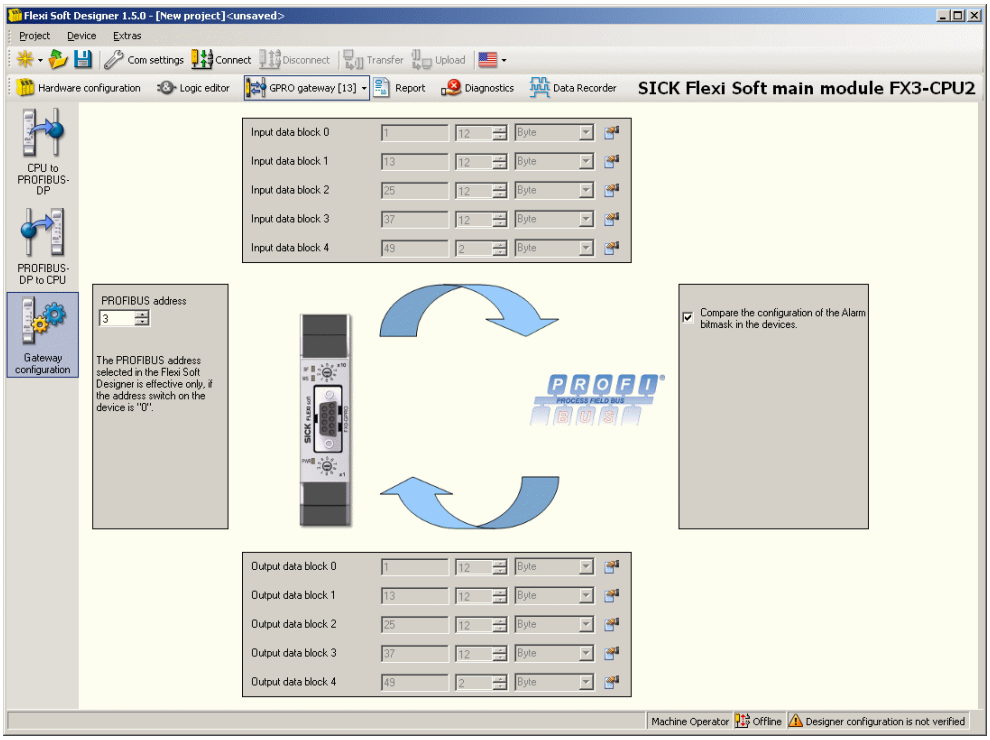

Figure 34: Configuration window for the PROFIBUS DP gateway

► Click on the button to the right of the required data block. The following dialog box is displayed:

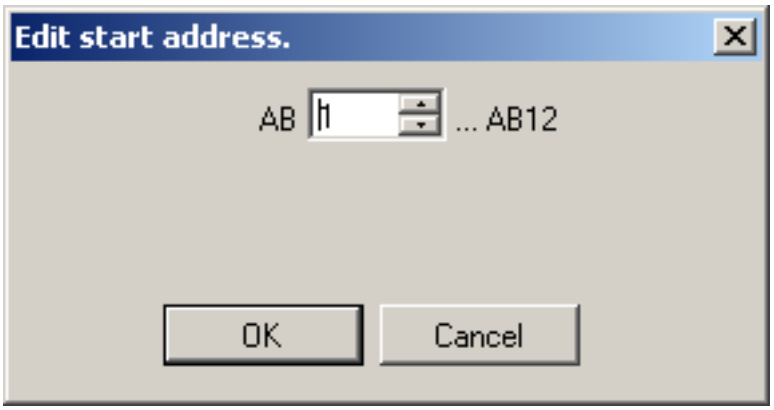

Figure 35: Editing data block start address

► Enter the desired new start address or change the start address using the arrows. The set address is automatically checked for plausibility (i.e., it is not possible to configure data blocks with overlapping address ranges).
► Click on **OK** to apply the new start address.

Information on how to configure the process image: see "Process image configuration", page 140 and the operating instructions "Flexi Soft in the Flexi Soft Designer configuration software" (SICK part number 8012998).

FX0-GPRO PROFIBUS DP diagnostic data

With the FX0-GPRO, diagnostic data is 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. The gateway determines the manufacturer-specific diagnostic number based on this ID. In this way, it is possible to read out module-specific diagnostic texts from the GSD. The content of the diagnostic messages is shown in the following table.

Table 61: Content of the PROFIBUS diagnostic messages

Octet	Content	Comment
7	09h	Header
8	Flexi Soft module number	The module number (see below) of the Flexi Soft module that caused the diagnostic message
9	0	PROFIBUS slot number of the module. The PROFIBUS DP gateway supports five slots. However, these do not represent physical slots and all messages must therefore be assigned to slot 0 (the gateway itself).
10 (bits 0 ... 2)	000, 001 or 010	000 = All errors outgoing, 001 = Incoming error, 010 = Outgoing error
10 (bits 3 ... 7)	00000 ... 11111	Alarm sequence number, is incremented with each change of state of octet 10, bit 0 ... 2 (error incoming/outgoing) The alarm sequence number is not used for modules with firmware V1.30.0 and higher 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 that caused the diagnostic message 0 = Main module 1 = 1st expansion module ... 13 = 1st gateway 14 = 2nd gateway (relay modules are not counted.)
12 ... 15	Variable	4 bytes with module-specific diagnostic data (see below)

The following table shows the module numbers of the Flexi Soft system.

Table 62: Flexi Soft module numbers

Module number	Module number [Hex]	Module
161	A1h	Flexi Soft main module (FX3-CPUx)
162	A2h	Expansion modules (FX3-XTIO, FX3-XTDI)
163	A3h	PROFIBUS DP gateway (FX0-GPRO)
164	A4h	CANopen gateway (FX0-GCAN)
165	A5h	DeviceNet gateway (FX0-GDEV)
166	A6h	Modbus TCP gateway (FX0-GMOD)
167	A7h	EtherNet/IP™ gateway (FX0-GENT)
168	A8h	PROFINET IO gateway (FX0-GPNT)
172	ACh	CC-Link gateway (on request)
175	AFh	SERCOS III gateway (on request)
176	B0h	EtherCAT gateway (FX0-GETC)
179	B3h	EFI-pro gateway (FX3-GEPR)
192	C0h	Expansion module (FX0-STIO)
193	C1h	Motion control (FX3-MOCx)
194	C2h	Expansion module (FX3-XTDS)

Module number	Module number [Hex]	Module
195	C3h	Analog input module (FX3-ANA0)

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

Table 63: PROFIBUS error messages

Module number [hex]	Diagnostic bit [Octet Bit]	Cause of error	Error message
1	12.0	Main module	Reserved
	12.1		Internal error: internal tests failed.
	12.2		External error
	12.3		Reserved
	12.4		Configuration status of a module in the system is incompatible or invalid.
	12.5		Voltage supply outside of the specified range
	12.6		EFI1 communication error
	12.7		EFI2 communication error
	13.0 ... 15.7		Reserved

Module number [hex]	Diagnostic bit [Octet Bit]	Cause of error	Error message
2	12.0	FX3-XTIO/ FX3-XTDI	Reserved
	12.1		Internal error: internal tests failed.
	12.2		External error
	12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5		Voltage supply outside of the specified range
	12.6 ... 12.7		Reserved
	13.0		Input 1–2 dual-channel evaluation: error
	13.1		Input 3–4 dual-channel evaluation: error
	13.2		Input 5–6 dual-channel evaluation: error
	13.3		Input 7–8 dual-channel evaluation: error
	13.4 ... 13.7		Reserved
	14.0		Input 1 external error for test signal. Check for stuck-at-high or wiring error.
	14.1		Input 2 external error for test signal. Check for stuck-at-high or cabling error.
	14.2		Input 3 external error for test signal. Check for stuck-at-high or cabling error.
	14.3		Input 4 external error for test signal. Check for stuck-at-high or cabling error.
	14.4		Input 5 external error for test signal. Check for stuck-at-high or cabling error.
	14.5		Input 6 external error for test signal. Check for stuck-at-high or cabling error.
	14.6		Input 7 external error for test signal. Check for stuck-at-high or cabling error.
	14.7		Input 8 external error for test signal. Check for stuck-at-high or cabling error.
	15.0		Output 1 test evaluation stuck-at-high error.
	15.1		Output 1 test evaluation stuck-at-low error.
	15.2		Output 2 test evaluation stuck-at-high error.
	15.3		Output 2 test evaluation stuck-at-low error.
	15.4		Output 3 test evaluation stuck-at-high error.
	15.5		Output 3 test evaluation stuck-at-low error.
	15.6		Output 4 test evaluation stuck-at-high error.
	15.7		Output 4 test evaluation stuck-at-low error.
3	12.0	PROFIBUS-DP gateway	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved

Module number [hex]	Diagnostic bit [Octet Bit]	Cause of error	Error message
4	12.0	CANopen gateway	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved
5	12.0	DeviceNet gateway	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved
6	12.0	Modbus TCP gateway	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved
7	12.0	EtherNet/IP ™ gateway	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved
8	12.0	PROFINET IO gateway	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved
0C	12.0	CC-Link gateway	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved
0F	12.0	Sercos III gateway	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved
10	12.0	EtherCAT gateway	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved

Module number [hex]	Diagnostic bit [Octet Bit]	Cause of error	Error message
13	12.0	EFI-pro gateway	Reserved
	12.1		Internal error: Internal tests have failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved
14 ... 1F	12.0 ... 15.7	Other gateways	Reserved
20	12.0	FX0-STIO	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5		Voltage supply outside of the specified range
	12.6		Reserved
	12.7		Output load (overcurrent) monitoring
	13.0 ... 15.7		Reserved

Module number [hex]	Diagnostic bit [Octet Bit]	Cause of error	Error message
21	12.0	FX3-MOCx	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5		Reserved
	12.6		Encoder 1 status
	12.7		Encoder 2 status
	13.0		Encoder 1 teach status
	13.1		Encoder 2 teach status
	13.2 ... 13.3		Reserved
	13.4		User-defined status bit 1 [1)]
	13.5		User-defined status bit 2 [1)]
	13.6		User-defined status bit 3 [1)]
	13.7		User-defined status bit 4 [1)]
	14.0		User-defined monitoring bit 1
	14.1		User-defined monitoring bit 2
	14.2		User-defined monitoring bit 3
	14.3		User-defined monitoring bit 4
	14.4		User-defined monitoring bit 5
	14.5		User-defined monitoring bit 6
	14.6		User-defined monitoring bit 7
	14.7		User-defined monitoring bit 8
	15.0		User-defined monitoring bit 9
	15.1		User-defined monitoring bit 10
	15.2		User-defined monitoring bit 11
	15.3		User-defined monitoring bit 12
	15.4		User-defined monitoring bit 13
	15.5		User-defined monitoring bit 14
	15.6		User-defined monitoring bit 15
	15.7		User-defined monitoring bit 16

Module number [hex]	Diagnostic bit [Octet Bit]	Cause of error	Error message
22	12.0	FX3-XTDS	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5		Power supply of the outputs
	12.6		Reserved
	12.7		Output load (overcurrent) monitoring
	13.0		Input 1–2 dual-channel evaluation: error detected
	13.1		Input 3–4 dual-channel evaluation: error detected
	13.2		Input 5–6 dual-channel evaluation: error detected
	13.3		Input 7–8 dual-channel evaluation: error detected
	13.4 ... 13.7		Reserved
	14.0		External test signal Input 1: error
	14.1		External test signal Input 2: error
	14.2		External test signal Input 3: error
	14.3		External test signal Input 4: error
	14.4		External test signal Input 5: error
	14.5		External test signal Input 6: error
	14.6		External test signal Input 7: error
	14.7		External test signal Input 8: error
	15.0 ... 15.7		Reserved
23	12.0	FX3-ANA0	Reserved
	12.1		Internal error: internal tests failed.
	12.2 ... 12.3		Reserved
	12.4		Configuration is incompatible or invalid.
	12.5 ... 15.7		Reserved
24 ... 3F	12.0 ... 15.7	Other modules	Reserved

1) The status of this bit can be defined for a specific application in the FX3-MOCx logic, e.g. in order to indicate prohibited movements of an axis that were detected by an FX3-MOCx function block.

4.2.2 FXO-GCAN CANopen gateway

4.2.2.1 Setting CANopen address

Overview

The CANopen address and the data transmission rate can be set using the switches on the gateway or by using the configuration software. Also see the operating instructions entitled “Flexi Soft Gateway Hardware” (SICK part number 8012662).

Important information

NOTE

- The hardware address switches can be used to set an address ranging from 1 to 99.
- With the configuration software, an address can be set ranging from 1 to 127.
- The CANopen master is not able to overwrite the address.
- If you use the configuration software to set the CANopen address and the data transmission rate, the settings will take effect as soon as the configuration is transferred (i.e., without having to switch the Flexi Soft system off and on first).
- Exception: If the system is in the busoff state, the device has to be switched off and back on again for the address change to take effect.

Procedure

Setting the address and data transmission rate using the configuration software

1. Set the address switch on the front of the device to “00”.
2. Start Flexi Soft Designer and load the hardware configuration, including the CANopen gateway. Then disconnect Flexi Soft Designer from the Flexi Soft system.
3. Click on **Interfaces** above the main window and select the FX0-GCAN or double-click on the FX0-GCAN in the **Hardware configuration** to open the dialog box for gateway configuration.
4. Click on **Gateway configuration**. The following dialog box is displayed:

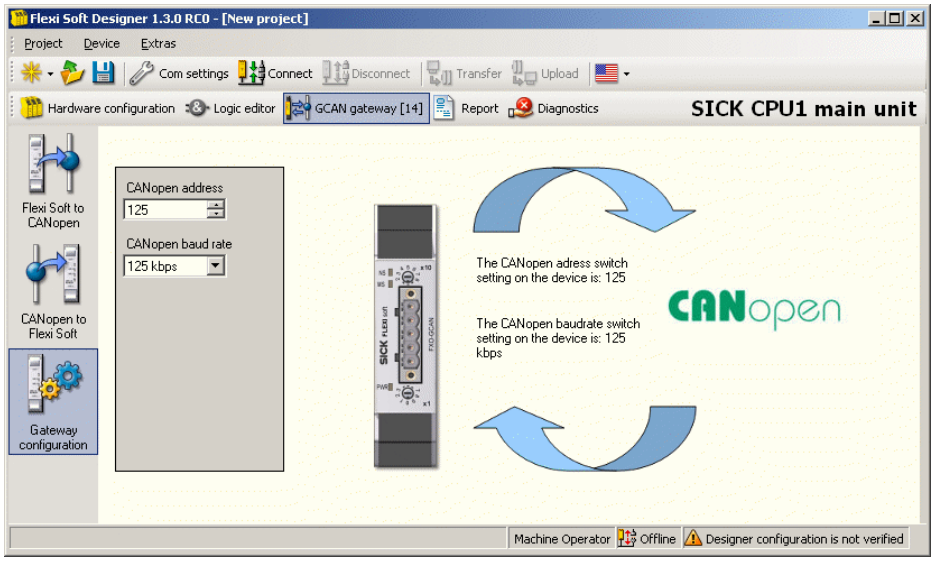

Figure 36: Setting the CANopen address of the FX0-GCAN

5. Select the CANopen address in the **CANopen address** field.
6. Select the data transmission rate in the **CANopen baud rate** field.
7. Click on **Connect** to switch to online mode.
8. Click on **Transfer** to transfer the configuration to the Flexi Soft system.

4.2.2.2 Selection of data

Each CANopen device stores its data in objects, which are listed in the object directory. The service data objects (SDOs) primarily contain the CANopen configuration data, while the process data is stored in process data objects (PDOs). Communication objects are used to read and write these SDOs and PDOs and to control the devices.

4.2.2.2.1 PCS – predefined connection set

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 consists of 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 consisting of a function code and a device address. The device address for the broadcast objects is 0, that of the other objects is in the range from 1 to 127.

Table 64: Structure of the CAN identifier

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

Table 65: PCS communication objects

Object	CAN-ID	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 (remote transmission request) bit, followed by the data length code (DLC), followed by 0 to 8 data bytes. The DLC (4 bits) indicates the number of data bytes.

4.2.2.2.2 NMT – Network management

NMT – Network management

The NMT broadcast object is used to start, stop or initialize CANopen devices. A device in the CANopen network must assume the role of NMT controller for this purpose. This is typically the PLC. All other devices are considered NMT devices. NMT services are broadcast services to which the devices do not generate responses.

All NMT objects start with the CAN ID 00h.

Broadcast service for an NMT device with address N

Table 66: Network management for an NMT device with address N

CAN-ID	DLC	Data							
00h	2	OP	N						

Broadcast service for all NMT devices

Table 67: Network management for all NMT devices

CAN-ID	DLC	Data							
00h	2	OP	0						

OP	NMT command	Definition
80h	Change to Pre-Operational	After booting, an NMT device automatically goes into the pre-operational state. In this state, communication via SDOs is allowed but not via PDOs. The NMT device can be switched from another state into this state.
01h	Change to Operational	The operational state is reached from the pre-operational state. In this state, communication via PDOs is possible and the CANopen device responds to SYNC commands. **Note:** When transitioning to the operational NMT state, every device sends a TxPDO with transmission type = 255 to ensure the NMT controller is informed of the current input configuration.
02h	Change to Prepared/Stopped	Communication via SDO or PDO is not possible in this state and the device also does not react to SYNC commands.
81h	Change to Reset Node	Initiates a reinitialization of the CANopen functionality in the NMT device.
82h	Change to Reset Communication	Initiates a reinitialization of the CANopen functionality in the NMT device; the toggle bit for node guarding is set to 0.

Example for resetting the entire communication

The following NMT object (CAN ID = 00h) contains 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):

Table 68: Example NMT object for resetting all communication

CAN-ID	DLC	Data							
00h	2	82h	0						

4.2.2.2.3 SYNC

SYNC

The SYNC command causes all TxPDOs of a CANopen device to be sent. It is therefore possible to poll the device using SYNC.

Table 69: Polling inputs using SYNC

CAN-ID	DLC	Data							
80h	0								

The device sends all input values when it receives this command. All TxPDOs are sent.

In order to ensure that the device automatically sends the current input values when it receives a SYNC command, the transmission type for the affected PDOs must be set to 1 (cyclic, synchronous). The device must also be in the operational state.

Further topics

- table 96

4.2.2.2.4 Emergency

Emergency

A CANopen device with the address N sends an emergency message to inform the other devices of an error state.

Table 70: 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 position of the module causing the error in the Flexi Soft system. The module position is displayed in the configuration software. The main module is always at position 0, the gateways are always at positions 13 and 14.								
M2 ... M5	4 bytes of module-specific status bits. Active bits have the value 1.								

The diagnostic bits for M2 to M5 are assigned as follows:

Table 71: CANopen Emergency, diagnostic bits M2 to M5

Bit 0	Bit 1	...	Bit 7	Bit 8	...	Bit 31
M5.0	M5.1	...	M5.7	M4.0	...	M2.7

Module-specific emergency status bits and messages

The status bits have the following meaning unless otherwise specified:

- 0 = Error
- 1 = No error

Table 72: Main module emergency messages

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	EFI2 communication error	EFI1 communication error	Voltage supply outside of the specified range	Configuration status of a module in the system is incompatible or invalid.	Reserved	Summary of bits 0.5 to 3.7 (external error)	Internal error: Internal tests have failed.	Reserved
Byte 1 ... 3	Reserved							

Table 73: Emergency messages FX3-XTIO/FX3-XTDI

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Reserved		Voltage supply outside of the specified range	Configuration is incompatible or invalid.	Reserved	Summary of bits 0.5 to 3.7 (external error)	Internal error: Internal tests have failed.	Reserved
Byte 1	Reserved				Input 7–8 dual-channel evaluation: error	Input 5–6 dual-channel evaluation: error	Input 3–4 dual-channel evaluation: error	Input 1–2 dual-channel evaluation: error

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 2	Input 8 external error for test signal. Check for stuck-at-high or cabling error.	Input 7 external error for test signal. Check for stuck-at-high or cabling error.	Input 6 external error for test signal. Check for stuck-at-high or cabling error.	Input 5 external error for test signal. Check for stuck-at-high or cabling error.	Input 4 external error for test signal. Check for stuck-at-high or cabling error.	Input 3 external error for test signal. Check for stuck-at-high or cabling error.	Input 2 external error for test signal. Check for stuck-at-high or cabling error.	Input 1 external error for test signal. Check for stuck-at-high or cabling error.
Byte 3	Output 4 test evaluation stuck-at-low error	Output 4 test evaluation stuck-at-high error	Output 3 test evaluation stuck-at-low error	Output 3 test evaluation stuck-at-high error	Output 2 test evaluation stuck-at-low error	Output 2 test evaluation stuck-at-high error	Output 1 test evaluation stuck-at-low error	Output 1 test evaluation stuck-at-high error

Table 74: Gateway emergency messages

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Reserved	Communication into the network. FX3-GEPR only: Bit 6 corresponds to the behavior of the NS LED.	Reserved	Configuration is incompatible or invalid.	Reserved		Internal error: Internal tests have failed.	Reserved
Byte 1 ... 3	Reserved							

Table 75: Emergency messages FX0-STIO

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Output load (overcurrent) monitoring	Reserved	Voltage supply outside of the specified range	Configuration is incompatible or invalid.	Reserved	Summary of bits 0.5 to 3.7 (external error)	Internal error: Internal tests have failed.	Reserved
Byte 1 ... 3	Reserved							

Table 76: Emergency messages FX3-MOCx

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Encoder 2 status	Encoder 1 status	Reserved	Configuration is incompatible or invalid.	Reserved	Summary of bits 0.5 to 1.7 (external error)	Internal error: Internal tests have failed.	Operational status of the module (1 = Run, 0 = Other)
Byte 1	User-defined status bit 4 [1)]	User-defined status bit 3 [1)]	User-defined status bit 2 [1)]	User-defined status bit 1 [1)]	Reserved		Encoder 2 teach status	Encoder 1 teach status
Byte 2	User-defined monitor bit 8	User-defined monitor bit 7	User-defined monitor bit 6	User-defined monitor bit 5	User-defined monitor bit 4	User-defined monitor bit 3	User-defined monitor bit 2	User-defined monitor bit 1
Byte 3	User-defined monitor bit 16	User-defined monitor bit 15	User-defined monitor bit 14	User-defined monitor bit 13	User-defined monitor bit 12	User-defined monitor bit 11	User-defined monitor bit 10	User-defined monitor bit 9

1) The status of this bit can be defined in the FX3-MOCx logic to suit the application, e.g., to indicate impermissible movements of an axis that were detected by an FX3-MOCx function block.

Table 77: Emergency messages FX3-XTDS

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Output load (overcurrent) monitoring	Reserved	Voltage supply outside of the specified range	Configuration is incompatible or invalid.	Reserved	Summary of bits 0.5 to 3.7 (external error)	Internal error: Internal tests have failed.	Reserved
Byte 1	Reserved				Input 7–8 dual-channel evaluation: error detected.	Input 5–6 dual-channel evaluation: error detected.	Input 3–4 dual-channel evaluation: error detected.	Input 1–2 dual-channel evaluation: error detected.
Byte 2	Input 8 external test signal: error	Input 7 external test signal: error	Input 6 external test signal: error	Input 5 external test signal: error	Input 4 external test signal: error	Input 3 external test signal: error	Input 2 external test signal: error	Input 1 external test signal: error
Byte 3	Reserved							

Table 78: Emergency messages FX3-ANA0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Sensor AI2 lower input range undercut	Sensor AI1 upper input range exceeded	Sensor AI1 lower input range undercut	Configuration is incompatible or invalid.	Reserved		Internal error: Internal tests have failed.	Reserved
Byte 1	Upper process range exceeded	Lower process range undercut	Discrepancy error	Sensor AI2 upper process range exceeded	Sensor AI2 lower process range undercut	Sensor AI1 upper process range exceeded	Sensor AI1 lower process range undercut	Sensor AI2 upper input range exceeded
Byte 2 ... 3	Reserved							

4.2.2.2.5 Node guarding

Node guarding

An NMT controller (e.g., a PLC with an integrated CANopen controller) uses the NMT error control object to detect the failure of an NMT device with the address N. The NMT device must respond to the NMT controller request within the node guarding time. The node guarding time must be monitored by the NMT controller.

The NMT controller sends a CAN message with the identifier 700h + node ID and RTR bit (remote transmission request).

Table 79: NMT controller request

CAN-ID	RTR	DLC	Data							
700h + N	1	0								

The device (e.g., the FX0-GCAN) then sends the byte 1 status byte with the following content:

Table 80: Device response

CAN-ID	DLC	Data							
700h + N	1	Byte1							

Table 81: Status byte Byte 1

Bit	Significance
7	Toggle bit, changes value between two sequential requests

Bit	Significance	
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 1017h contains a value for the heartbeat producer time, see table 91, page 105), it sends a cyclic message with the CAN ID 700h+N, DLC = 1 and Byte 1 = 05h. The toggle bit (bit 7 of byte 1) is always 0.

Heartbeat consumer

If the gateway is configured as a heartbeat consumer (i.e., if SDO 1016.1h contains a value for the heartbeat consumer time, see table 91, page 105), then at least one node guarding message must be received within the configured heartbeat consumer time (typically from an NMT controller).

4.2.2.2.6 PDO communication

PDO communication

Process Data Objects (PDOs) are the real-time objects of the CANopen fieldbus. They are sent without protocol overhead, i.e. the receiver does not send any confirmation.

The FX0-GCAN provides four Transmit Process Data Objects (TxPDOs) containing the operating data that is to be sent into the network and four Receive Process Data Objects (RxPDOs) for the operating data received from the network.

CANopen objects are addressed via 11-bit CAN identifiers. As the default, the CAN identifier of each object is derived from the object type and configured CANopen device address. The CAN identifiers of the PDOs can be changed with the SDOs 1400h to 1403h for the RxPDOs and the SDOs 1800h to 1803h for the TxPDOs ("PDO linking").

NOTE

- Each Process Data Object contains 8 bytes.
- The default content of the Process Data Objects can be changed using the configuration software.

Table 82: Default content of the Transmit Process Data Objects (TxPDOs) of the FX0-GCAN

	PDO1	PDO2	PDO3	PDO4
	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	Direct gateway output values 1
Byte 5	Input values, module 2	Input values, module 10	Output values, module 6	Direct gateway output values 2
Byte 6	Input values, module 3	Input values, module 11	Output values, module 7	Direct gateway output values 3
Byte 7	Input values, module 4	Input values, module 12	Output values, module 8	Direct gateway output values 4

NOTE

- The process data can also be written and read using the Service Data Objects SDO 6000h and SDO 6200h. Simple access via SDO is recommended for diagnostic purposes. The faster PDO communication should be used in normal operation.
- After startup or a configuration change (either via the CANopen master or via the configuration software), the MS LED of the CANopen gateways flashes Red/green until a first transmit/receive data exchange by means of PDO or SDO 6000h/SDO 6200h has taken place in the CANopen network.

TxPDO1 ... 4

A transmit PDO transmits data from the CANopen gateway to a CANopen device.

Table 83: TxPDO1 ... 4

CAN-ID	DLC	Data							
181h ... 1FFh	8	B1	B2	B3	B4	B5	B6	B7	B8
281h ... 2FFh	8	B9	B10	B11	B12	B13	B14	B15	B16
381h ... 3FFh	8	B17	B18	B19	B20	B21	B22	B23	B24
481h ... 4FFh	8	B25	B26	B27	B28	B29	B30	B31	B32

B1 ... B32: CAN telegram bytes as mapped in the network input data.

The gateway sends one or more TxPDOs if at least one of the following events occurs:

- At last one input or output byte has changed its value and the transmission type for the TxPDO containing this byte has the value 255.
- At last 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, then 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 1800.5h for TxPDO1) has a value n > 0. In this case, this TxPDO is sent every n ms.
- A TxPDO can also be polled via a 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.

The operational status of the device must be **Operational** for all transmission types (see table 67, page 98).

RxPDO1 ... 4

A receive PDO transmits data from a CANopen device to the CANopen gateway.

Table 84: RxPDO1 ... 4

CAN-ID	DLC	Data							
201h ... 1FFh	8	B1	B2	B3	B4	B5	B6	B7	B8
301h ... 2FFh	8	B9	B10	B11	B12	B13	B14	B15	B16
401h ... 3FFh	8	B17	B18	B19	B20	B21	B22	B23	B24
501h ... 4FFh	8	B25	B26	B27	B28	B29	B30	B31	B32

B1 ... B32: CAN telegram bytes as shown in the gateway input data.

Transmission type 255 is the default for all RxPDOs. This means that the gateway immediately forwards the received RxPDO data to the main module. This setting cannot be changed.

4.2.2.2.7 SDO communication

SDO communication

SDOs are Service Data Objects. These objects contain a wide range of different data. This also includes configuration data as well as input and output data. Unlike with PDO communication, reception of each SDO is answered on protocol level, i.e. the receiving device sends a confirmation.

The following protocols are supported in this CANopen PCS implementation:

- SDO Download Expedited (write SDO)
- SDO Upload Expedited (read SDO)
- Upload SDO Segment Protocol (segmented reading of an SDO)

SDO Download Expedited (write SDO)

The client sends a request to server N. The 16-bit index and the subindex for the SDO to be written are contained in this message. The request additionally contains four data bytes with the data to be written.

Table 85: Write SDO

CAN-ID	DLC	Data							
600h + N	8	23h	SDO_L	SDO_H	SUB	Byte 1	Byte 2	Byte 3	Byte 4

SDO_L = SDO index, low byte

SDO_H = SDO index, high byte

SUB = SDO subindex

The server then replies with a confirmation message:

Table 86: SDO write confirmation

CAN-ID	DLC	Data							
580h + N	8	60h	SDO_L	SDO_H	SUB	Byte 1	Byte 2	Byte 3	Byte 4

Bytes 1 to 4 in the write confirmation message contain zeros.

SDO Upload Expedited (read SDO)

The client requests the content of an SDOs with a request to server N. The 16-bit index and the subindex for the SDO to be read are contained in this message. Bytes 1 to 4 in the read request message contain zeros.

Table 87: 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 replies with the following message: Bytes 1 to 4 contain the value of the requested object.

Table 88: SDO read 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

To transmit the data types UDINT or UINT, the data must be in Intel format or little endian format. For example, the 32-bit value 12345678h in the data bytes 5, 6, 7 and 8 must be transmitted in the following sequence: [5] = 78, [6] = 56, [7] = 34, [8] = 12.

NOTE
This also applies to the SDO index in data bytes 2 and 3, which has the data type UINT. This means that the low byte is transmitted in data byte 2 and the high byte in data byte 3.

Example: The following messages are required in order to read SDO 1003.1h of the CANopen device with the device address 2. The data type of the data to be read is UDINT.

The client sends:

Table 89: Read SDO (example)

CAN-ID	DLC	Data							
602h	8	40h	03h	10h	01h	00h	00h	00h	00h

The server responds:

Table 90: SDO read conformation (example)

CAN-ID	DLC	Data							
582h	8	43h	03h	10h	01h	08h	00h	50h	02h

The response data combined produce the 32-bit word 02500008h.

4.2.2.2.8 SDO object directory

SDO object directory

Each CANopen device manages its SDOs in an object directory. The complete object directory is formally described in an EDS file. Many CANopen tools can read this EDS file and as a result know the object characteristics of the CANopen device.

The following table shows all SDOs of the FX0-GCAN gateway.

Table 91: Supported SDOs

SDO	Type
1000h	Device type

SDO	Type
1001h	Error register
1003h	Error list (error history)
1005h	COB ID SYNC
1008h	Device name
1009h	Hardware version
100Ah	Software version
100Ch	Guard time
100Dh	Life time factor
1014h	COB ID EMGY (available from version V1.30.0)
1016h	Heartbeat consumer time
1017h	Heartbeat producer time
1018h	Identification
1027h	Module list
1400h ... 1403h	Communication parameters for RxPDO1 ... 4
1600h ... 1603h	Mapping parameters for RxPDO1 ... 4
1800h ... 1803h	Communication parameters for TxPDO1 ... 4
1A00h ... 1A03h	Mapping parameters for TxPDO1 ... 4
3100h	Module status bits
3200h	Config CRC
3300h	Module type code
6000h	Process data input objects
6200h	Process data output objects

Detailed information on these SDOs is provided in the CANopen draft standard DS 301 V4.02 (DSP 301 V4.1).

SDO 1001h: Error register

The error register (SINT) contains an error bit that indicates whether an error is present. A "generic error" has been detected if bit 0 is set to 1.

SDO 1003h: Error list (error history)

SDO 1003h is an array that contains the last 10 error codes reported by the gateway via emergency messages. Array index 0 contains the number of error codes that have been recorded in SDO 1003h.

A new error is recorded in index 1, and older errors are then renumbered in this case (incremented by 1). The array index can be overwritten externally with a 0, which will clear the array completely.

NOTE

- Not all errors that are reported via emergency messages are recorded in SDO 1003h, rather only the listed errors: "Emergency", page 99.
- The entries in SDO 1003h are in UDINT format and are normally divided into 16 bits of error code and 16 bits of additional information. The module status diagnostics (4 bytes) are entered here in the case of an emergency message.

SDO 1005h: COB ID SYNC

SDO 1005h contains the COB ID of the SYNC object. This value is 80h as default, but can be changed.

NOTE
If the COB ID of the SYNC object is to be changed, it must be ensured that the new COB ID is not already assigned to another communication object.

SDO 1008h: Device name

SDO 1008h contains a device name (VISIBLE STRING).

NOTE
This SDO cannot 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 1009h: Hardware version

SDO 1009h contains the current hardware version of the device (VISIBLE STRING).

NOTE
This SDO cannot 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 100Ah: Software version

SDO 100Ah contains the current software version of the device (VISIBLE STRING).

NOTE
This SDO cannot 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 100Ch: 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 reports a cable break error and sets all process data from the network to 0; the NS LED starts to flash Red.

Life guarding is activated in the slave by the first node guarding message if the set life guarding time is not 0. If the guard time or life time factor is set to 0 after life guarding has been activated, life guarding will be deactivated (see "Guarding protocols", page 113).

SDO 100Dh: Life time factor

SDO 100Dh contains the life time factor (SINT), see SDO 100Ch.

NOTE
The life time factor must be either = 0 (deactivated) or ≥ 1.5.

SDO 1016h: Heartbeat consumer time

The gateway is configured as the heartbeat consumer if SDO 1016h contains a value greater than 0 for the heartbeat consumer time. The heartbeat consumer time is given in milliseconds.

The NMT master must send at least one node guarding message to the slave within this time. If the heartbeat consumer time is exceeded (life guarding error), the gateway reports a cable break error and sets all process data from the network to 0; the NS LED starts to flash Red.

SDO 1017h: Heartbeat producer time

The gateway can also function as a heartbeat producer, i.e. send a heartbeat signal. This allows another device to recognize whether the heartbeat producer (i.e. the gateway) is still functioning correctly.

The heartbeat producer time is given in milliseconds. It is rounded up to the next highest multiple of four for internal processing. 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 "Guarding protocols", page 113).

SDO 1018h: Identification

This SDO contains basic information about the gateway.

Table 92: Content of SDO 1018h

Sub index	Mapping	Format	Description
1	Manufacturer ID	UDINT	Unique ID number 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 1027h: Module list

The module list contain the module type and module ID of all Flexi Soft modules in the system.

Table 93: Content of SDO 1027h

Sub index	Module	Format
1	Main module	SINT
2 ... 13	Expansion modules	SINT
14, 15	Gateways	SINT

Module types and module IDs: see "Emergency", page 99. The value for free module slots is 0.

SDO 1400h ... 1403h: Communication parameters for RxPDO1 to RxPDO4

The communication parameters for RxPDO1 to RxPDO4 can be configured with SDO 1400h to 1403h. For example, SDO 1400h defines the parameters for RxPDO1 etc.

Table 94: Content of SDO 1400h ... 1403h

Sub index	Mapping	Format	Description
1	COB-ID	UDINT	CAN identifier for this PDO, write-protected
2	Receive mode	SINT	Fix 255 (asynchronous mode)

The receive mode (read/write) determines how the PDO shall be received. Receive mode is set to 255 (asynchronous mode) for the RxPDOs. In this mode, the data of a received RxPDO is routed directly to the outputs.

NOTE

If receive mode is set to a value other than 255, an error code will be generated (abort code 0609 0030h, invalid parameter value).

SDO 1600h ... 1603h: Mapping parameters for RxPDO1 to RxPDO4

This SDO cannot be used because mapping of the RxPDOs takes place using the configuration software (see table 84, page 104).

SDO 1800h ... 1803h: Communication parameters for TxPDO1 to TxPDO4

The communication parameters for TxPDO1 to TxPDO4 can be configured with SDO 1800h to 1803h. For example, SDO 1800h defines the parameters for TxPDO1 etc.

Table 95: Content of SDO 1800h ... 1803h

Sub index	Mapping	Format	Description
1	COB-ID	UDINT	CAN identifier for this PDO, write-protected
2	Transmission type	SINT	Determines when the PDO is to be sent
5	Event timer	UINT	In milliseconds

As default, the transmission type of all TxPDOs is set to 255 (asynchronous mode, event-controlled).

The event timer contains the time in milliseconds for the cyclic transmission of the TxPDOs.

Transmission types for the TxPDOs

Table 96: Transmission types for the TxPDOs

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 will be generated (abort code 0030 0030h, invalid parameter value).

Synchronous: Synchronous transmission type 0 means that the TxPDO will be sent after a SYNC command is received, but only if data has changed. The synchronous transmission types n = 1 ... 240 define that the TxPDO will be sent after the n-th SYNC command is received.

Asynchronous, event-controlled in the event of a change of state and by the timer: The asynchronous transmission type 254/255 means that the TxPDO is sent every time a change is made to at least one input bit included in this PDO or when the event timer has elapsed. For example, if the event timer has a value of 500, this means that the gateway sends the respective TxPDO every 500 ms.

RTR, on request: The transmission type 253 means that the TxPDO can be requested by an RTR (Remote Transmission Request). This requires a CAN message to the gateway with DLC = 0, RTR = 1 and the COB ID of the TxPDO. The gateway then replies with the requested TxPDO.

SDO 1A00h ... 1A03h: Mapping parameters for the TxPDOs

This SDO cannot be used because mapping of the TxPDOs takes place using the configuration software (see table 82, page 103 and see table 83, page 103).

SDO 3100h: Module status bits

SDO 3100h contains the module status bits of the Flexi Soft system (see "Emergency", page 99). Active bits have the value 0.

Table 97: Content of SDO 3100h

SDO array	Data set parameters	Module	Size
3100h, byte 1	Status Module 0	Main module	UDINT
3100h, byte 2	Status Module 1	Expansion	UDINT
...	...	...	...
3100h, byte 14	Status Module 13	Gateway	UDINT
3100h, byte 15	Status Module 14	Gateway	UDINT

NOTE
The positions of the modules are numbered from 0 to 14 in the configuration software. The subindex for SDO 3100h is therefore = module position + 1.

SDO 3100h can only be read.

SDO 3200h: Config CRC

SDO 3200h contains the Flexi Soft checksums in big endian format (inverted UDINT format).

SDO 3300h: Module type code

SDO 3300h contains the type codes of the max. 15 modules in the Flexi Soft system in SINT format (8 bytes per module = 120 bytes).

Table 98: Module type code in SDO 3300h

Byte	Bit	Value	Code	Designation
0	0 ... 3	**System**		
		07h	FX	Flexi Soft safety controller
	4 ... 7	**Safety integrity level**		
		00h	0	None
		01h	1	SIL1
		02h	2	SIL2
		03h	3	SIL3

Byte	Bit	Value	Code	Designation
1	0 ... 7	**Module type**		
		00h	FX3-CPU0	Main module of the Flexi Soft safety controller
		01h	FX3-CPU1	Main module of the Flexi Soft safety controller with EFI
		02h	FX3-CPU2	Main module of the Flexi Soft safety controller with EFI and ACR
		03h	FX3-CPU3	Main module of the Flexi Soft safety controller with EFI, ACR, and Flexi Line
		04h	FX3-XTDI	Expansion module with safe inputs
		05h	FX3-XTDS	Expansion module with safe inputs and non-safe outputs
		06h	FX3-XTIO	Expansion module with safe inputs and safe outputs
		07h	FX0-GPRO	PROFIBUS-DP gateway
		08h	FX0-GDEV	DeviceNet gateway
		09h	FX0-GCAN	CANopen gateway
		0Ah	FX0-GENT	EtherNet/IP™ gateway
		0Bh	FX0-GMOD	Modbus TCP gateway
		0Ch	FX0-GPNT	PROFINET IO gateway
		14h	FX0-GCC1	CC-Link gateway
		15h	FX3-GS3S	Sercos III gateway
		16h	FX0-GETC	EtherCAT gateway
		18h	FX3-GEPR	EFI-pro gateway
		20h	FX0-STIO	Expansion module with non-safe inputs and non-safe outputs
		21h	FX3-MOC1	Motion control
		24h	FX3-MOC0	Motion control
		30h	FX3-ANA0	Analog input module
		FFh	Empty	No module type (empty configuration)
2 ... 6	0 ... 7	**For internal use**		

Byte	Bit	Value	Code	Designation
7	0 ... 7	**Module identification for diagnostic purposes**		
		00h	–	–
		01h	FX3-CPUxxxxxx	Main module of the Flexi Soft safety controller
		02h	FX3-XTDIxxxxx	Expansion module with safe inputs
			FX3-XTIOxxxxx	Expansion module with safe inputs and safe outputs
		03h	FX0-GPROxxxxx	PROFIBUS DP gateway
		04h	FX0-GCANxxxxx	CANopen gateway
		05h	FX0-GDEVxxxxx	DeviceNet gateway
		06h	FX0-GMODxxxxx	Modbus TCP gateway
		07h	FX0-GENTxxxxx	EtherNet/IP™ gateway
		08h	FX0-GPNTxxxxx	PROFINET IO gateway
		0Ch	FX0-GCC1xxxxx	CC-Link gateway
		0Fh	FX3-GS3Sxxxxx	SERCOS III gateway
		10h	FX0-GETCxxxxx	EtherCAT gateway
		13h	FX3-GEPRxxxxx	EFI-pro gateway
		20h	FX0-STIOxxxxx	Expansion module with non-safe inputs and non-safe outputs
		21h	FX3-MOCxxxxx	Motion control
		22h	FX3-XTDSxxxxx	Expansion module with safe inputs and non-safe outputs
		23h	FX3-ANA0xxxx	Analog input module

SDO 6000h: Process data input objects

The 32 bytes of process input data can be written to SDO array 6000h. This is the same data as in RxPDO1 to 4 (see table 84, page 104). The mapping is as follows:

Table 99: Mapping table for SDO 6000h – RxPDO1 ... 4

SDO 6000h	RxPDO
6000h, byte 1	RxPDO1 ... 4, Byte 1
...	...
6000h, byte 8	RxPDO1 ... 4, Byte 8
6000h, byte 9 ... 16	RxPDO2 ... 4, Byte 1 ... 8
6000h, byte 17 ... 24	RxPDO3 ... 4, Byte 1 ... 8
6000h, byte 25 ... 32	RxPDO4 ... 4, Byte 1 ... 8

SDO 6000h can only be written.

SDO 6200h: Process data output objects

The 32 bytes of process output data can be read from SDO array 6200h. This is the same data as in TxPDO1 to 4 (see table 83, page 103). The mapping is as follows:

Table 100: Mapping table for SDO 6200h – TxPDO1 ... 4

SDO 6200h	TxPDO
6200h, byte 1	TxPDO1, Byte 1
...	...
6200h, byte 8	TxPDO1, Byte 8
6200h, byte 9 ... 16	TxPDO2, Byte 1 ... 8

SDO 6200h	TxPDO
6200h, byte 17 ... 24	TxPDO3, Byte 1 ... 8
6200h, byte 25 ... 32	TxPDO4, Byte 1 ... 8

SDO 6200h can only be read.

4.2.2.3 Guarding protocols

CANopen offers several options for monitoring correct functioning of the fieldbus interface (e.g. cable break detection).

Guarding is obligatory according to CIA CANopen specification DS 301. Therefore, either node guarding or heartbeat must always be activated. If guarding is not configured, the Flexi Soft system cannot detect any interruption in CANopen communication, e.g., due to a broken network cable. In this case, the input and output data of the CANopen gateway may “freeze”.

NOTICE

Malfunction due to non-identified interruptions in CANopen communication

The input and output data of the CANopen gateway may no longer be updated if neglected.

► Always use node guarding or heartbeat.

Heartbeat

A heartbeat producer is a CANopen device that sends a cyclic heartbeat message. This allows all other CANopen devices to recognize whether the heartbeat producer is still functioning correctly and what its current status is. Heartbeat messages are sent at a regular time interval, the heartbeat producer time, which can be configured with SDO 1017h. The configured 16-bit value is rounded up to the next-higher multiple of 4 ms.

A heartbeat consumer is a CANopen device that expects a cyclic node guarding message within a certain time interval, the heartbeat consumer time, which can be configured with SDO 1016h. If the heartbeat consumer does not receive a node guarding message within the configured heartbeat consumer time, it sends a life guarding emergency message and sets the process input data to 0. The gateway additionally sends a "cable break" error message which can be processed by the main module.

Node guarding

Node guarding is performed by an NMT master. This can be any CANopen device that can perform this function as a client. The NMT master sends a cyclic node guarding message to the device to be monitored, which must then reply in a certain time that is monitored by the NMT master If the device to be monitored does not reply within the node guarding time, the NMT master treats this as a device malfunction and initiates the corresponding measures.

Life guarding

Life guarding is performed by the gateway itself. The life guarding time is calculated in the gateway from the values of SDO 100Ch (Guard Time) and SDO 100Dh (Life Time Factor). If the gateway does not receive a node guarding message from and NMT master at least once within this life guarding time, the gateway sends an internal "cable break" error message that can be processed by the main module and the LED NS starts to flash Red.

NOTE

- The gateway can detect a cable break if life guarding is activated, i.e. both SDO 100Ch and SDO 100Dh have a values that is not equal to 0. In this case, life guarding starts as soon as the first node guarding request is received from an NMT master and ends when the master sends the command Reset Communication.
- A cable break can also be detected if the gateway is configured as a heartbeat consumer. In this case, cable break detection is performed by the gateway itself.
- The heartbeat producer functions without node guarding. In this case, the gateway is not able to detect a cable break on the fieldbus.
- Heartbeat and node guarding/life guarding cannot be used simultaneously.
- If the configuration is changed so that life guarding is deactivated or activated, the entire Flexi Soft system must be restarted so that the CANopen network communication is established correctly again.

The following table provides an overview of the supported guarding protocols depending on the configuration of SDO 1016h and SDO 1017h (Heartbeat), SDO 100Ch (Guard Time) and SDO 100Dh (Life Time Factor).

Table 101: Overview and comparison of guarding protocols

SDO 1016h	SDO 1017h	SDO 100Ch × 100Dh	Heartbeat Gateway	Life Guarding Gateway	Node Guarding NMT master
0	0	0	Not permissible: Either node guarding or heartbeat must always be activated.		
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 permitted**		

4.2.2.4 Error objects

The FX0-GCAN reports CAN-specific errors (e.g. initialization errors, cable break, CAN communication errors) as FLEXBUS+ errors.

Module-specific errors (see "Emergency", page 99) are reported as extended diagnostics using the emergency object and SDO 1003h.

Emergency object

The emergency producer (CANopen gateway) sends the emergency object to the emergency consumer (any CANopen device, usually the control) if CAN-specific errors or an error state occur (see "Emergency", page 99).

The emergency object is sent as described in CANopen draft standard DS 301 (Section 9.2.5):

Table 102: Emergency states and transitions

Emergency state before	Transition	Module-specific alarms	Emergency state after
Error-free	1	Incoming error	Error occurred
Error occurred	2	Outgoing error, other errors present	Error occurred
Error occurred	3	Coming error, other errors present	Error occurred
Error occurred	4	All errors removed	Error-free

The gateway is in one of two possible emergency states, either Error-free or Error occurred. Emergency objects are sent between these two states depending on the transitions. The error code in the emergency object indicates the emergency state present in the gateway (see table 103, page 115).

Overview of error objects

Table 103: Overview of error objects

CAN-specific errors	Error code FLEXBUS+	Error type	Emergency error code Error register M1 ... M5	Error history SDO 1003h	Result/Possible remedy
CAN data overflow CAN control overflow in Rx Fifo	4501h	Warning	8110h 11h 1, 0, 0, 0, 0	–	CAN messages have been lost. Limited bandwidth Check CAN settings, increase data transmission rate, reduce number of nodes or data volume.
CAN error-passive CAN control is in error-passive state.	4503h	Warning	8120h 11h 0, 0, 0, 0, 0	–	The gateways is transmitting only recessive bits, i.e. it is making its own messages invalid. The cause is either a hardware fault on the gateway or an external data transmission fault. Check the cabling.
CAN bus off The CAN control is in busoff state.	4504h	Warning	–	–	Major transmission errors. The CAN control has disconnected the connection to the bus. Possible hardware fault. Switch the Flexi Soft system off and then back on again.
CAN Tx Fifo overflow The CAN control does not have any transmission resources.	4506h	Warning	8110h 11h 2, 0, 0, 0, 0	–	CAN messages that should have been sent by the gateway have been lost. The number of events for which the gateway should send CAN messages is too high for the set data transmission rate. Increase the data transmission rate or change the gateway configuration.
CAN initialization failed. The CAN control could not be initialized.	C507h	Serious	–	–	The CAN control or transceiver is possibly faulty. Replace the FX0-GCAN with a new device.

CAN-specific errors	Error code FLEX-BUS+	Error type	Emergency error code Error register M1 ... M5	Error history SDO 1003h	Result/Possible remedy
CANopen Life Guarding CANopen Life Guarding has detected a cable break	4508h	Warning	8130h 11h 0, 0, 0, 0, 0	–	The gateway has generated a life guarding error message. Either an error has occurred on the node guarding or heartbeat NMT master or the CAN cable is interrupted. Check the CANopen master. Check the cabling.
Module-specific alarms	**Error code FLEX-BUS+**	**Emergency state transition**	**Emergency error code Error register M1 ... M5**	**Error history SDO 1003h**	
Gateway detects incoming error in accordance with trigger conditions	–	1	FF01h 81h M1 = Module index M2 ... M5 = Module diagnostic data	M2, M3, M4, M5	"Emergency", page 99
Gateway detects outgoing error, other errors present	–	2	FF02h 81h M1 = Module index M2 ... M5 = Module diagnostic data	M2, M3, M4, M5	"Emergency", page 99
Gateway detects incoming error, other errors present	–	3	FF03h 81h M1 = Module index M2 ... M5 = Module diagnostic data	M2, M3, M4, M5	"Emergency", page 99
All errors removed	–	4	0000h 00h M1 = Module index M2 ... M5 = 0	–	

CANopen diagnostics examples

Example 1: FX3-XTIO module in position 1, output Q4 has short-circuit to High

The gateway sends an emergency message.

Table 104: Emergency message (example 1)

CAN-ID	DLC	Data							
08Ch	8	03h	FFh	01h	01h	40h	00h	00h	00h

The CANopen address of the gateway is 12 (hexadecimal = C). The FX3-XTIO module has position 1 in the Flexi Soft system.

08Ch: Identifier (80 + C)

8: Data length code: 8 bytes follow.

03h FFh: Error code FF03: device-specific error

01h: Error register 01 of SDO 1001h

01h: Module index M1: module at position 1

40h: Module status bit 30 (bit 6 of byte M2) = 1: short-circuit to High at output 4 (see "Emergency", page 99)

Reading the current error from SDO 3100h

PLC requests:

Table 105: Request from SDO 3100h via the PLC (example 1)

CAN-ID	DLC	Data							
60Ch	8	40h	00h	31h	02h	00h	00h	00h	00h

60Ch: Identifier (600 + C)

8: Data length code: 8 bytes follow.

40h: Expedited upload request

00h 31h: Index 3100

02h: Subindex: Module at position 1 (see table 97, page 110)

Response of the gateway:

Table 106: Response of the gateway from SDO 3100h (example 1)

CAN-ID	DLC	Data							
58Ch	8	42h	00h	31h	02h	BFh	FFh	FFh	FBh

58Ch: Identifier (580 + C)

8: Data length code: 8 bytes follow.

42h: Upload response, size of the data set is not displayed.

00h 31h: Index 3100

02h: Subindex: Module at position 1 (see table 97, page 110)

FBh: Error byte M5, bit 2 = 0: External error

BFh: Error byte M2, bit 30 = 0: Error: short-circuit to High at output 4

Reading the error from the error history in SDO 1003h

PLC requests:

Table 107: Request from SDO 1003h via the PLC (example 1)

CAN-ID	DLC	Data							
60Ch	8	40h	03h	10h	01h	00h	00h	00h	00h

60Ch: Identifier (600 + C)

8: Data length code: 8 bytes follow.

40h: Expedited upload request

03h 10h: Index 1003

01h: Subindex: last error

Response of the gateway:

Table 108: Response of the gateway from SDO 1003h (example 1)

CAN-ID	DLC	Data							
58Ch	8	42h	03h	10h	01h	40h	00h	00h	00h

58Ch: Identifier (580 + C)

8: Data length code: 8 bytes follow.

42h: Upload response, size of the data set is not displayed.

03h 10h: Index 1003

01h: Subindex: last error

40h: Module status bit 30 (bit 6 of byte M2) = 0: short-circuit to High at output 4

Example 2: FX3-XTDI module with error at the dual-channel input I1 / I2

The gateway sends an emergency message.

Table 109: Emergency message (example 2)

CAN-ID	DLC	Data							
08Ch	8	03h	FFh	01h	0Ch	00h	00h	01h	00h

The CANopen address of the gateway is 12 (hexadecimal = C). The FX3-XTDI module has position 12 in the Flexi Soft system.

08Ch: Identifier (80 + C)

8: Data length code: 8 bytes follow.

03h FFh: Error code FF03: device-specific error

01h: Error register 01 of SDO 1001h

0Ch: Module index M1: module at position 12 (hexadecimal = C)

01h: Module status bit 8 (bit 0 of byte M4) = 1: dual-channel evaluation of input 1 to 2: error detected (see "Emergency", page 99)

Reading the current error from SDO 3100h

PLC requests:

Table 110: Request from SDO 3100h via the PLC (example 2)

CAN-ID	DLC	Data							
60Ch	8	40h	00h	31h	0Dh	00h	00h	00h	00h

60Ch: Identifier (600 + C)

8: Data length code: 8 bytes follow.

40h: Expedited upload request

00h 31h: Index 3100

0Dh: Subindex 0D = module at position 12 (module position = subindex – 1, see table 97, page 110)

Response of the gateway:

Table 111: Response of the gateway from SDO 3100h (example 2)

CAN-ID	DLC	Data							
58Ch	8	42h	00h	31h	0Dh	FFh	FFh	FEh	FBh

58Ch: Identifier (580 + C)

8: Data length code: 8 bytes follow.

42h: Upload response, size of the data set is not displayed.

00h 31h: Index 3100

0Dh: Subindex 0D: module at position 12 (see table 97, page 110)

FBh: Error byte M5, bit 2 = 0: External error

FEh: Error byte M4, bit 0 = 0: Dual-channel evaluation of input 1 to 2: error detected (see "Emergency", page 99)

Reading the error from the error history in SDO 1003h

PLC requests:

Table 112: Request from SDO 1003h via the PLC (example 2)

CAN-ID	DLC	Data							
60Ch	8	40h	03h	10h	01h	00h	00h	00h	00h

60Ch: Identifier (600 + C)

8: Data length code: 8 bytes follow.

40h: Expedited upload request

03h 10h: Index 1003

01h: Subindex: last error

Response of the gateway:

Table 113: Response of the gateway from SDO 1003h (example 2)

CAN-ID	DLC	Data							
58Ch	8	42h	03h	10h	01h	00h	00h	01h	00h

58Ch: Identifier (580 + C)

8: Data length code: 8 bytes follow.

42h: Upload response, size of the data set is not displayed.

03h 10h: Index 1003

01h: Subindex: last error

01h: Module status bit 8 (bit 0 of byte M4) = 0: dual-channel evaluation of input 1 to 2: error detected

4.2.3 DeviceNet gateway FX0-GDEV

The following Flexi Soft gateway can be used for DeviceNet: FX0-GDEV.

4.2.3.1 Basic configuration – Setting the DeviceNet address

The DeviceNet address and the data transmission rate can be set either via the switches on the gateway or by using the configuration software. Also refer to the operating instructions "Flexi Soft Gateways Hardware" (SICK part number 8012662).

Setting the DeviceNet address and data transmission rate using the Flexi Soft Designer configuration software

- ► Set the address switch on the front of the device to “00”.
- ► Start Flexi Soft Designer and load the hardware configuration, including the DeviceNet gateway. Then disconnect Flexi Soft Designer from the Flexi Soft system.
- ► Click on the **Interfaces** button above the main window and select the FX0-GDEV or double-click on the FX0-GDEV in the **Hardware configuration** to open the dialog box for gateway configuration.
- ► Click on the **Gateway configuration** button. The following dialog box is displayed:

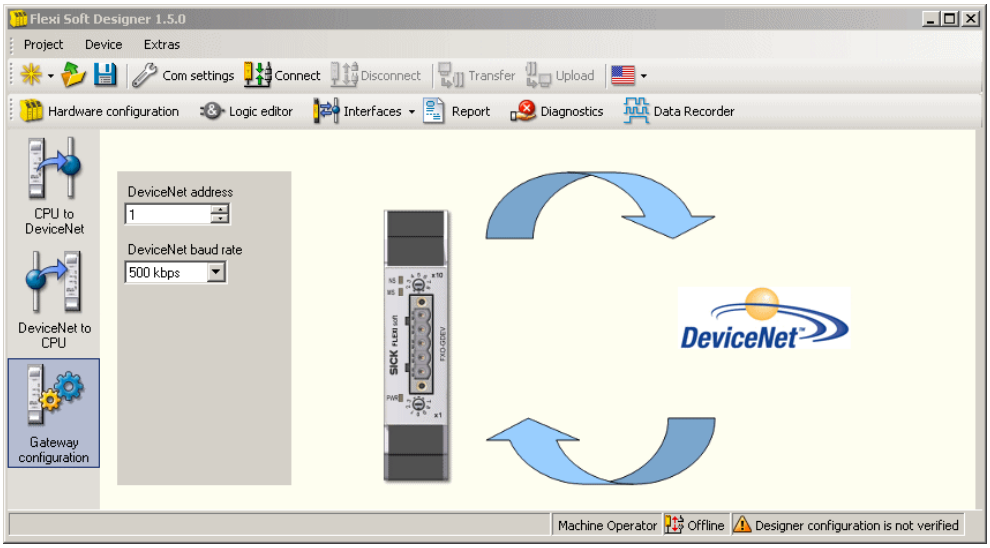

Figure 37: Setting the DeviceNet address of the FX0-GDEV

- ► Enter the DeviceNet address in the **DeviceNet address** field.
- ► Enter the data transmission rate in the **DeviceNet baud rate** field.
- ► Click on **Connect** to switch to online mode.
- ► Click on **Transfer** to transfer the configuration to the Flexi Soft system.

NOTE

- An address in the range 1 ... 63 can be set by means of the hardware address switches.
- An address in the range 0 ... 63 can be set by means of the configuration software.
- The DeviceNet master is not able to overwrite the address.
- If you use the configuration software to set the DeviceNet address and the data transmission rate, the settings will take effect as soon as the configuration is transferred (i.e. without having to switch the Flexi Soft system off and on first).
- Exception: If the system is in the “busoff” state, the device must be switched off and back on again for the change to take effect.

4.2.3.2 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)
- Group 2 Unconnected Explicit Request Messages (with predefined master/slave connection set)
- Device heartbeat messages and shutdown messages
- UCMM Port (Unconnected Message Manager)
- Duplicate MAC ID Messages
- Offline Connection Set

Predefined master/slave connection set

The FX0-GDEV supports a predefined master/slave connection set that allows a DeviceNet connection to be established that requires reduced network and device resources. This set contains an explicit messaging connection and allows 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 addresses a single, specific slave (point-to-point connection). A master must send a separate poll command to every slave to be polled. On receiving the poll command, the slave responds to the master with an I/O message.

Change of state I/O messages and cyclic I/O messages

A change of state I/O message is sent by either the master or the slave. Change of state I/O messages and cyclic I/O messages address a single, 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 reported with an explicit message response.

Device heartbeat messages and shutdown messages

The heartbeat message can be used in Change-Of-State mode (COS) in order to indicate to the master that the device is (still) in operation. The device sends a heartbeat message if no new COS message is generated for a configurable time.

A device can log out of the network with the optional shutdown message if it switches to offline state due to a fault or another reason.

Unconnected messages

Unconnected messages (UCMM messages) are used to establish or terminate explicit connections between two devices. They are processed by the Unconnected Message Manager (also known as the UCMM port). A maximum of three explicit connections can be established via UCMM at the same time.

The gateway accesses the PLC tags directly using an unconnected send message (service 0x52). This unconnected send message contains a message request with service code 0x4c (read tag) or 0x4d (write tag). In the EPATH of the message request, an ANSI extended symbol segment is used to address the PLC tag directly via its ASCII-coded name.

Group 2 Only Unconnected Explicit Request Messages

Group 2 Only Unconnected Explicit Request Messages are used to assign or terminate the predefined master/slave connection set. They occur only for devices that do not support a UCMM port and serve as an alternative for establishing a connection. However, only one connection with a single partner is possible, namely the master.

Duplicate MAC ID messages

These messages are used to call the Network Access State Machine, which prevents two or more nodes from having an identical MAC ID in the same network.

4.2.3.3 DeviceNet protocol settings

Supported Service Command Codes

Table 114: Supported Service Command Codes

Service code	Service name
0Eh	Get_Attribute_Single
10h	Set_Attribute_Single
05h	Reset command

Assembly Object settings

Table 115: Assembly Object settings

Parameter name	Significance	Value
SUPPORT_ASSEMBLY_ATTRIB_1N2	0 = Attributes 1 and 2 of the Assembly Object are not supported. 1 = 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. 1 = Attribute 4 of the Assembly Object is supported.	1
ASMOBJ_NUM_OF_INSTANCES	Number of assembly instances	22

4.2.3.4 Assembly Objects

Assembly Objects are used to exchange input and output data that consist of more than one attribute via a single connection. A data package is produced that can be referenced as attribute 3 of the Assembly Object class (Class 4).

Produced Assembly Instance (Target ➜ Originator)

Table 116: Produced Assembly Instance

Instance ID	Description	Size [bytes]	Received data
1	Input data sets 1 ... 4	50 82 142 202	Input data set 1 Input data sets 1 ... 2 Input data sets 1 ... 3 Input data sets 1 ... 4
2	Input data sets 2 ... 4	32 92 152	Input data set 2 Input data sets 2 ... 3 Input data sets 2 ... 4
3	Input data sets 3 ... 4	60 120	Input data set 3 Input data sets 3 ... 4
4	Input data set 4	60	Input data set 4
10	Input data set 1	50	Input data set 1
11	Input data set 2	32	Input data set 2
12	Input data set 3	60	Input data set 3
13	Input data sets 1 ... 3	50 82 142	Input data set 1 Input data sets 1 ... 2 Input data sets 1 ... 3
18	Input data set 1, block 1	10	Input data set 1, bytes 0 ... 9
19	Input data set 1, block 2	10	Input data set 1, bytes 10 ... 19
20	Input data set 1, block 3	10	Input data set 1, bytes 20 ... 29
21	Input data set 1, block 4	10	Input data set 1, bytes 30 ... 39
22	Input data set 1, block 5	10	Input data set 1, bytes 40 ... 49

NOTE

- All values are of the data type **Array of USINT**. The possible values are therefore in the range 0 ... 255.
- The access mode for all instances is **GET**.
- All Assembly Objects can be accessed via implicit messaging or explicit messaging.

Consumed Assembly Instance (Originator ➜ Target)

Table 117: Consumed Assembly Instance

Instance ID	Description	Size [bytes]	Sent data
5	Output data sets 1 ... 5	10 20 30 40 50	Output data set 1 Output data sets 1 ... 2 Output data sets 1 ... 3 Output data sets 1 ... 4 Output data sets 1 ... 5
6	Output data sets 2 ... 5	10 20 30 40	Output data set 2 Output data sets 2 ... 3 Output data sets 2 ... 4 Output data sets 2 ... 5
7	Output data sets 3 ... 5	10 20 30	Output data set 3 Output data sets 3 ... 4 Output data sets 3 ... 5
8	Output data sets 4 ... 5	10 20	Output data set 4 Output data sets 4 ... 5
9	Output data set 5	10	Output data set 5
14	Output data set 1	10	Output data set 1
15	Output data set 2	10	Output data set 2
16	Output data set 3	10	Output data set 3
17	Output data set 4	10	Output data set 4

NOTE

- All values are of the data type **Array of USINT**. The possible values are therefore in the range 0 ... 255.
- The access mode for all instances is **GET/SET**.
- All Assembly Objects can be accessed via implicit messaging or explicit messaging.

Object definition for Full Data Set Transfer (72h – one instance)

The manufacturer-specific object **Full Data Set Transfer** defines the attributes with which the PLC can perform the following actions:

- request complete input data sets from the FX0-GDEV
- write complete output data sets to the FX0-GDEV

Class attributes (instance 0)

Table 118: Class attributes for the object Full Data Set Transfer (72h)

Attribute ID	Name	Data type	Data value(s)	Access type
1	Revision	UINT	1	Get
2	Max. instance	UINT	1	Get
3	Number of instances	UINT	1	Get

Instance attributes

Table 119: Instance attributes for the object Full Data Set Transfer (72h) (instance 1)

Attribute ID	Name	Data type	Data value(s)	Access type
Flexi Soft to the network				
1	Request of input data set 1-specific data	UINT	0 ... 255	Read
2	Request input data set 2-specific data	UINT	0 ... 255	Read
3	Request input data set 3-specific data	UINT	0 ... 255	Read
4	Request input data set 4-specific data	UINT	0 ... 255	Read
Network to Flexi Soft				
5	Write output data set 1-specific data	UINT	0 ... 255	Write
6	Write output data set 2-specific data	UINT	0 ... 255	Write
7	Write output data set 3-specific data	UINT	0 ... 255	Write
8	Write output data set 4-specific data	UINT	0 ... 255	Write
9	Write output data set 5-specific data	UINT	0 ... 255	Write
Number of assemblies and size of the input and output data sets				
33	Size of the output data sets	USINT	0 ... 50	Write
34	Size of the input data sets	USINT	0 ... 202	Write
35	Consumed Assembly Instance	USINT	5 ... 9, 14 ... 17	Write
36	Produced Assembly Instance	USINT	1 ... 4, 10 ... 13, 18 ... 22	Write

Common services

Table 120: Common services for the object Full Data Set Transfer (72h)

Service code	Implemented in class	Implemented in instance	Service name
0Eh	Yes	Yes	Get_Attribute_Single
10h	No	Yes	Set_Attribute_Single

Definition of instance attributes (instance 1)

These attributes offer access to entire input and output data sets. The **Get Attribute Single** command requests a specific input data set and only returns the parameter information for the requested data set. The **Set Attribute Single** command describes a specific output data set.

Object Individual Input Data Set Transfer (73h – one instance per data set)

The manufacturer-specific object **Individual Input Data Set Transfer** defines the attributes with which the PLC can request individual parameters within a data set.

Class attributes

Table 121: Class attributes for the object Individual Input Data Set Transfer (73h)

Attribute ID	Name	Data type	Data value(s)	Access type
1	Revision	UINT	1	Read
2	Max. instance	UINT	4	Read
3	Number of instances	UINT	4	Read

Instance attributes

Table 122: Instance attributes for the object Individual Input Data Set Transfer (73h)

Attribute ID	Name	Data type	Data value(s)	Access type
1 to 4 (depending on the definition of the data set)	Request input data set-specific data	Depending on the definition of the data set	0 ... 255	Read

Common services

Table 123: Common services for the object Individual Data Set Transfer (73h)

Service code	Implemented in class	Implemented in instance	Service name
0Eh	Yes	Yes	Get_Attribute_Single

Definitions of instance attributes

Attributes 1 to 4 – Request input data-specific parameters
These attributes return the input data set-specific data arrays.

Get Attribute Single – Requests for a specific input data set return only the parameter information for the requested data set.

The attributes numbered consecutively 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 show the attribute definitions for each input data set.

All data set information is returned in little endian format.

Example:

The data is returned as follows for an input data set:

- IntegerArray[0]: BBAAh – AA = BYTE1; BB = BYTE2
- IntegerArray[1]: DDCCh – CC = BYTE3; DD = BYTE4
- ...
- IntegerArray[6]: NNMMh – MM = BYTE13; NN = BYTE14

Instance 1 – Attribute definitions for input data set 1

Table 124: Attribute definitions for instance 1 of object Individual Input Data Set Transfer (73h)

Attribute ID	Data set parameters	Size
1	Byte 0	SINT
2	Byte 1	SINT
...	...	...
50	Byte 49	SINT

Instance 2 – Attribute definitions for input data set 2

Table 125: Attribute definitions for instance 2 of object Individual Input Data Set Transfer (73h)

Attribute ID	Data set parameters	Size
1	Overall checksum	UDINT
2	Flexi Soft checksum	UDINT
3	FX3-CPU0 and FX3-CPU1: reserved FX3-CPU2 and FX3-CPU3: ACR checksum	UDINT
4	Flexi Soft checksum (verified)	UDINT
5	Reserved	UDINT
6	Reserved	UDINT
7	Reserved	UDINT
8	Reserved	UDINT

Instance 3 – Attribute definitions for input data set 3

Table 126: Attribute definitions for instance 3 of object Individual Input Data Set Transfer (73h)

Attribute ID	Data set parameters	Size
1	Status Module 0	UINT[2]
2	Status Module 1	UINT[2]
...	...	...
15	Status Module 14	UINT[2]

Instance 4 – Attribute definitions for input data set 4

Table 127: Attribute definitions for instance 4 of object Individual Input Data Set Transfer (73h)

Attribute ID	Data set parameters	Size
1	Reserved	UINT[2]
2	Reserved	UINT[2]
...	...	...
15	Reserved	UINT[2]

5 The process image

5.1 Data transferred to the network (network input data sets)

Available data

The Flexi Soft gateways can make the following data available:

- Operating data
  - **Input values** (1/0) of all Flexi Soft input expansion modules in the system as well as the connected EFI-enabled devices (see "Direct gateway output values", page 130)
  - **Output values** (1/0) of all Flexi Soft expansion modules as well as the connected EFI-enabled devices (see "Direct gateway output values", page 130)
  - **Logic results** of the Flexi Soft main module (FX3-CPUx) (see "Logic results", page 130)
  - **Output data** from another network, i.e. data that were received from a second gateway in the Flexi Soft system (see "Routing data from a second network", page 132)
- Diagnostics
  - **Error and status information** for all modules except for UE410-2RO and UE410-4RO (see "Error and status information of the modules", page 133)
  - **Checksums** (see "Configuration checksums", page 133)

Data sets

The physical Flexi Soft modules are not represented as typical hardware modules in the network. Instead, the data provided by the Flexi Soft system is organized in four input data sets.

- **Data set 1** (max. 50 bytes) contains the operating data. The content of data set 1 can be modified using Flexi Soft Designer. It is preset when the gateway is delivered (for details, see table 129, page 129).
  With the FX0-GPNT and FX0-GPRO, data set 1 is split into five input data blocks, whereby data blocks 1 to 4 each contain 12 bytes and data block 5 contains two bytes. The FX0-GCAN contains four process data objects, each with eight bytes. For detailed information, please refer to the section of these operating instructions that corresponds to the gateway in question.
- **Data set 2** (32 bytes) contains the checksums of the system configuration (see "Configuration checksums", page 133).
- **Data set 3** (60 bytes) contains the status and diagnostic data of the individual modules with four bytes per module in each case (details: see "Error and status information of the modules", page 133).
- **Data set 4** (60 bytes) is currently filled with reserved values and should therefore not be used for the application.

The following table provides an overview of which data sets can be made available by which gateway.

Table 128: Availability of data sets 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	–	– [1)]	–

	Data set 1	Data set 2	Data set 3	Data set 4
FX0-GCAN	CANopen	CANopen (SDOs)	CANopen (SDOs) [2]	–
FX0-GDEV	DeviceNet	DeviceNet	DeviceNet	DeviceNet
FX0-GETC	Input and output data objects 2000h and 2001h	Checksums Object 2002h	Status and diagnostics Object 2003h	Reserved Object 2004h

1) The FX0-GPRO makes status and diagnostic data available via PROFIBUS standard DP-V0 diagnostics see table 61, page 89.

2) The FX0-GCAN makes status and diagnostic data available via CANopen SDOs (Service Data Objects), see "SDO communication", page 104.

Table 129: Overview of input data sets 1–3 (default setting for EtherNet/IP™, Modbus TCP, and TCP/IP)

	Data set 1	Data set 2	Data set 3
Byte 0	Logic result 0	Overall checksum	Status Module 0. Module 0 is always the main module. Detailed information on the module status: see "Error and status information of the modules", page 133.
Byte 1	Logic result 1		
Byte 2	Logic result 2		
Byte 3	Logic result 3		
Byte 4	Input values, module 1	Flexi Soft checksum	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	FX3-CPU0 and FX3-CPU1: reserved FX3-CPU2 and FX3-CPU3: ACR checksum	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	Flexi Soft checksum (verified)	Status Module 3
Byte 13	Input values, module 10		
Byte 14	Input values, module 11		
Byte 15	Input values, module 12		

	Data set 1	Data set 2	Data set 3
Byte 16	Output values, module 1	Reserved	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		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		Status Module 6
Byte 25	Output values, module 10		
Byte 26	Output values, module 11		
Byte 27	Output values, module 12		
Byte 28	Direct gateway output values 1		Status Module 7
Byte 29	Direct gateway output values 2		
Byte 30	Direct gateway output values 3		
Byte 31	Direct gateway output values 4		
Byte 32 ... 49	Not assigned	Not available	Status Module 8–13
Byte 50 ... 55	Not available		
Byte 56			Status Module 14 (Modules 13 and 14 are always the gateways)
Byte 57			
Byte 58			
Byte 59			
Length	50 bytes	32 bytes	60 bytes

5.1.1 Logic results

The logic results generated by the Flexi Soft main module logic editor can be transferred to the network. Up to 20 bytes are available for this, whereby each bit represents a logic result from the logic editor. Data set 1, which contains the logic results, can be adjusted as needed. For detailed information, see "Process image configuration", page 140 and the chapter on the gateway in question.

5.1.2 Direct gateway output values

It is possible to write values directly from the logic editor to a gateway. For this purpose, four data bytes are reserved as default for data set 1; however, all 50 data bytes of data set 1 can be used as gateway output values. Additional information: see "Direct gateway output values", page 142.

NOTE
A main module with firmware version V2.00.0 or higher is needed in order to used direct gateway output values.

5.1.3 Module and EFI status and input and output values

The Flexi Soft gateways can transfer the status plus the input and output values of all Flexi Soft modules and EFI-enabled devices connected to the Flexi Soft system to the network. Data set 1, which contains the input and output values plus the EFI information, can be adjusted as needed. For detailed information, see "Process image configuration", page 140 and the chapter on the gateway in question.

Module status

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

Table 130: Module status

Module status	Size	Significance	Assignment
Input data status	2 bytes	One sum bit per module for the status of the module inputs 0 = Error 1 = No error	Bit 0 = 1st expansion module Bit 1 = 2nd expansion module ... Bit 12 = 1st gateway Bit 13 = 2nd gateway Bit 14 = Reserved Bit 15 = Reserved
Output data status	2 bytes	One sum bit per module for the status of the module outputs 0 = Error 1 = No error	
Position status	2 bytes	One sum bit per module for the status of the module inputs and outputs (AND link of input data status and output data status) 0 = Error 1 = No error	Bit 0 = Main module Bit 1 = 1st expansion module Bit 2 = 2nd expansion module ... Bit 13 = 1st gateway Bit 14 = 2nd gateway Bit 15 = Reserved

Detailed information on the significance of the status bits is contained in the operating instructions "Flexi Soft in the Flexi Soft Designer configuration software" (SICK part number 8012998).

NOTE
The input and output statuses of the FX3-XTIO and FX3-XTDI modules are available as from firmware version V2.00.0.

Input and output values of the modules

- Input values for I/O modules:
  For each module, one byte is available for data set 1. The input values show the status of the preliminary evaluation on the I/O module. This corresponds to the status of the element in the logic of the main module. The level at the associated terminal is not guaranteed to be identified from this as the data can be set to 0 due to the cross-circuit detection or dual-channel evaluation, regardless of the level at the input terminal (e.g., I1 to 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 of 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 (bits 1, 3, 5 and 7) is used as follows in this case:

Table 131: Use of the more significant bit for two-channel evaluation on I/O modules FX3-XTIO, FX3-XTDI or FX3-XTDS

Firmware version FX3-XTIO, FX3-XTDI or FX3-XTDS	Dual-channel equivalent switches	Dual-channel complementary switches
V1.xx.x	Same status as the less significant bit	Inverted status of the less significant bit
V2.00.0 and higher	Status of pre-evaluation 0 = Error 1 = No error	

- Output values for I/O modules:
  1 byte is available for data set 1 in each case for each module with outputs.
  The output values indicate the status of the control information from the logic of the main module for the related element on the I/O module The level at the related terminals cannot be reliably detected by this means, as the output may be deactivated by the cross circuit detection or overload protection.
  If dual-channel output elements are configured at an I/O module, then only the low-order 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). In this case, the high-order bit (bit 1, 3, 5, and 7) is not used, i.e., it has the value 0.
- Input values for FX3-MOCx modules:
  2 bytes are available for data set 1 in each case for each FX3-MOCx module. The input values indicate the status of the signals from the logic of the main module to the FX3-MOCx logic. Bit 16 and bit 17 of the usable bits from the main module to the FX3-MOCx logic are not available here.
- Output values for FX3-MOCx modules:
  2 bytes are available for data set 1 in each case for each FX3-MOCx module. The output values indicate the status of the signals from the FX3-MOCx logic to the logic of the main module.

EFI system information

The main modules from FX3-CPU1 have two EFI interfaces. An EFI interface is a secure communication interface between EFI-capable SICK devices. It allows ...

- information to be read out from the EFI-capable safety devices (e.g. C4000, S3000).
- commands to be transmitted to the EFI-capable safety devices.

The Flexi Soft gateways can transmit the data of the EFI-capable devices connected to the main module into the network.

NOTE
EFI data can only be selected in byte arrays. 4 byte arrays are available for each connected EFI-capable device. Some of the data content is reserved and cannot be used in the PLC.

For more detailed information about the properties, functions, and advantages of the enhanced function interfaces, please see the operating instructions titled “Flexi Soft Modular Safety Controller Hardware” (SICK part number 8012999).

The general EFI function description is contained in the Technical Information "EFI – Enhanced Function Interface" (SICK part number 8012621).

5.1.4 Routing data from a second network

If a Flexi Soft system contains two gateways, it is possible to forward information that the first gateway receives from a network (e.g., from a Modbus PLC) to a second network (e.g., to a PROFINET master) via the second gateway and vice versa.

5.1.5 Configuration checksums

Data set 2 contains the following configuration checksums of the Flexi Soft system.

- Overall checksum:
  If ACR is deactivated: the same value as the Flexi Soft checksum
  If ACR is activated: checksum over Flexi Soft checksum and ACR checksum
- Flexi Soft checksum:
  This checksum covers the configuration for the Flexi Soft system, i.e., for all Flexi Soft modules. If safety-related changes are made to the configuration (e.g., adding a safety-related device), the Flexi Soft checksum changes. The configuration for any connected EFI-enabled devices is not included in the Flexi Soft checksum.
- Flexi Soft checksum (verified):
  This is the Flexi Soft checksum at the time of the most recent verification. For the purpose of verification, the Flexi Soft checksum of the current configuration is used in the Designer and compared with the checksum of the configuration stored in the device. If these two checksums are identical, the configuration of the Flexi Soft system is considered to be verified (LED CV lights up ● Yellow).
- ACR checksum:
  This checksum covers the ACR configuration for EFI-enabled devices.

Each checksum is four bytes long. Data set 2 cannot be changed.

5.1.6 Error and status information of the modules

Overview

Data set 3 contains the status information of the modules that is transmitted into the network.

Four bytes are transferred for each module. Data set 3 cannot be adjusted.

The module status bits have the following meaning unless otherwise specified:

- 0 = Error
- 1 = No error

Important information

NOTE

- 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 of each module are transferred as a 32-bit word in big endian format, i.e., the most significant byte (MSB = byte 3) is transferred first and the least significant byte (LSB = byte 0) last.

Module status bits of the main modules

Table 132: Module status bits of the main modules

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	EFI2	EFI1	Voltage supply	Configuration of the Flexi Soft system	Reserved	Summary of bits 0.5 to 3.7 (external error)	Internal tests	Operational status of the module 1 = Run 0 = Other

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Reserved					Status of Flexi Line 1 = No error 0 = Error	Flexi Link stations suspended 1 = None 0 = One or more	Flexi Link stations in the system 1 = All found 0 = One or more missing
Byte 2 ... 3	Reserved							

Module status bits of the FX3-XTIO and FX3-XTDI I/O modules

The module status bits for the FX3-XTIO and the FX3-XTDI are only fully supported for firmware version V1.20.0 and higher.

Table 133: Module status bits of the FX3-XTIO and FX3-XTDI I/O modules

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Reserved	Fast shut off output	Voltage supply of the outputs	Configuration of this module is valid.	Reserved	Summary of bits 0.5 to 3.7 (external error)	Internal tests	Operational status of the module 1 = Run 0 = Other
Byte 1	Reserved				Status I7, I8 Dual-channel evaluation	Status I5, I6 Dual-channel evaluation	Status I3, I4 Dual-channel evaluation	Status I1, I2 Dual-channel evaluation
Byte 2	External test signal for input 8	External test signal for input 7	External test signal for input 6	External test signal for input 5	External test signal for input 4	External test signal for input 3	External test signal for input 2	External test signal for input 1
Byte 3	Short circuit monitoring for output 4 Short-circuit to Low	Short circuit monitoring for output 4 Short-circuit to High	Short circuit monitoring for output 3 Short-circuit to Low	Short circuit monitoring for output 3 Short-circuit to High	Short circuit monitoring for output 2 Short-circuit to Low	Short circuit monitoring for output 2 Short-circuit to High	Short circuit monitoring for output 1 Short-circuit to Low	Short circuit monitoring for output 1 Short-circuit to High

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

Table 134: Module status bits of the I/O module FX3-XTDS

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Output load (overcurrent) monitoring	Reserved	Voltage supply of the outputs	Configuration of this module is valid.	Reserved	Summary of bits 0.5 to 3.7 (external error)	Internal tests	Operational status of the module 1 = Run 0 = Other
Byte 1	Reserved				Status I7, I8 Dual-channel evaluation	Status I5, I6 Dual-channel evaluation	Status I3, I4 Dual-channel evaluation	Status I1, I2 Dual-channel evaluation
Byte 2	External test signal for input 8	External test signal for input 7	External test signal for input 6	External test signal for input 5	External test signal for input 4	External test signal for input 3	External test signal for input 2	External test signal for input 1
Byte 3	Reserved							

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

Table 135: Module status bits of the FX0-STIO I/O module

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Output load (overcurrent) monitoring	Reserved	Voltage supply of the outputs	Configuration of this module is valid.	Reserved	Summary of bits 0.5 to 3.7 (external error)	Internal tests	Operational status of the module 1 = Run 0 = Other
Byte 1 ... 3	Reserved							

Module status bits of the FX3-ANA0 analog input module

Table 136: Module status bits of the FX3-ANA0 analog input module

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Sensor AI2 lower input range	Sensor AI1 upper input range	Sensor AI1 lower input range	Configuration of this module is valid.	Reserved	Reserved	Internal tests	Reserved
Byte 1	Upper process range limit	Lower process range limit	Discrepancy status	Sensor AI2, upper process range	Sensor AI2, lower process range	Sensor AI1, upper process range	Sensor AI1, lower process range	Sensor AI2 upper input range
Byte 2 ... 3	Reserved							

Module status bits of the FX3-MOCx motion control module

Table 137: Module status bits of the FX3-MOCx motion control module

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Encoder 2 is OK	Encoder 1 is OK	Reserved	Configuration of this module is valid.	Reserved	Summary of bits 0.5 to 1.7 (external error)	Internal tests	Operational status of the module 1 = Run 0 = Other
Byte 1	User-defined MOC status bit 4 [1]	User-defined MOC status bit 3 [1]	User-defined MOC status bit 2 [1]	User-defined MOC status bit 1 [1]	Reserved		Teach position for encoder 2 is OK	Teach position for encoder 1 is OK
Byte 2	User-defined MOC monitor bit 8	User-defined MOC monitor bit 7	User-defined MOC monitor bit 6	User-defined MOC monitor bit 5	User-defined MOC monitor bit 4	User-defined MOC monitor bit 3	User-defined MOC monitor bit 2	User-defined MOC monitor bit 1
Byte 3	User-defined MOC monitor bit 16	User-defined MOC monitor bit 15	User-defined MOC monitor bit 14	User-defined MOC monitor bit 13	User-defined MOC monitor bit 12	User-defined MOC monitor bit 11	User-defined MOC monitor bit 10	User-defined MOC monitor bit 9

[1] The status of this bit can be defined in the FX3-MOCx logic to suit the application, e.g., to indicate impermissible movements of an axis that were detected by an FX3-MOCx function block.

Module status bits of the gateway

Table 138: Module status bits of the gateway

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Reserved	Communication into the network. FX3-GEPR only: Bit 6 corresponds to the behavior of the NS LED.	Communication from the network. FX3-GEPR only: Bit 5 is always 1 (master/originator mode).	Configuration of this module is valid.	Reserved	Summary of bits 0.5 to 3.7 (external error)	Internal tests	Operational status of the module 1 = Run 0 = Other
Byte 1 ... 3	Reserved							

Example

Module 2 (FX3-XTIO) has a short-circuit to High (24 V) at output 3. The following module status is transferred to the network (only the first 12 bytes of 60 are displayed):

Table 139: 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**	...
Significance	Status Module 0 (main module)				Status Module 1 (FX3-XTIO)				**Status Module 2 (FX3-XTIO)**				...

The first relevant byte for the error on module 2 described above is module status byte 0 for module 2. This is byte 11 of data set 3 with the hexadecimal value FB (11111011):

Table 140: 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: see table 133, page 134).

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

Table 141: 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: see table 133, page 134).

Example process image: see "Example of a TCP/IP process image", page 82.

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

The data received from the network is organized in output data sets (max. 50 bytes). In the case of FX0-GENT, FX0-GMOD, FX0-GPNT, FX0-GETC and FX0-GDEV, these data sets are divided into five data blocks of 10 bytes each. In the case of the FX0-GPROI, the output data blocks 1 to 4 contain 12 bytes each and output data block 5 contains 2 bytes. The FX0-GCAN has four process data objects with 8 bytes each.

Table 142: Output data blocks 1–5 of the gateways

Gateway	Size of output data block 1	Size of output data block 2	Size of output data block 3	Size of output data block 4	Size of output data block 5
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-GDEV	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	–

The content of the output data sets can be used in the logic editor of the main module and can also be made available for another network via a second gateway in the Flexi Soft system.

NOTE

- In order to make it possible to use the data from the network in the logic editor or another network, every bit that is to be used must be assigned a tag name.
- Bits without a specific tag name are not available in the logic editor for routing via a second gateway. The corresponding section on the respective gateways contains detailed information on how to assign tag names for the received data.
- The communication status with the network can be monitored in the logic editor. Status bits are available for this purpose for data reception from the network and transmission into the network. If a gateway detects an error in communication, both the content of the data sets and the corresponding status bit will be set to zero (logical 0).
- In case all communication is dropped, the data of the output data sets and the input data status bit will be set to zero (logical 0).
- If one connection is closed while others are still available, the MS LED or the STATUS LED on the affected gateway will flash Red/green for 10 seconds and an entry will be made in the error history. The status bits will not be affected in this case.

NOTE

Use different output data set numbers for different PLC connections or TCP/IP sockets.

The output data set of the Ethernet gateway can be changed simultaneously via several communication interfaces or TCP/IP sockets (e.g. Modbus TCP and Ethernet TCP/IP) if the same output data set number is used for the different connections. In this case, the last message overwrites the data received earlier. To prevent this, a output data set number must be used for each connection.

5.3 Routing

The process image transferred from the Flexi Soft gateway to the network is made up of the operating data (e.g., logic results, status of the inputs and outputs) and the diagnostic data (e.g., module status, checksums). This data is split into four data sets.

Table 143: Content of data sets 1 to 4

Data set	Content	Size	Configurable
1	Operating data	Max. 50 bytes [1]	Yes
2	Checksums	32 bytes	No
3	Status and diagnostics	60 bytes	No
4	Reserved	60 bytes	No

1) FX0-GCAN: 32 bytes.

The operating data in data set 1 is divided into one or more data blocks depending on the network protocol. Detailed information on modularization of the data that is sent into the network: see table 144, page 139 and the section on the respective gateway.

The content of data set 1 is preconfigured in the delivery status, but can also be freely configured as required (see "Default settings for the operating data", page 138 and see "Process image configuration", page 140).

The diagnostic data in data sets 2 to 4 depends on the network protocol used and is described in the section on the respective gateway.

5.4 Default settings for the operating data

The operating data is default on delivery. Depending on the gateway used, this data is divided into several data blocks.

The following table provides an overview of the assigned bytes in the default configuration and also shows the modularization of the data for the different gateways.

Table 144: Default configuration for the operating data transmitted to 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)	Inputs, module 1	1 (12 bytes)
1	Logic result 1		Inputs, module 2	
2	Logic result 2		Inputs, module 3	
3	Logic result 3		Inputs, module 4	
4	Inputs, module 1		Inputs, module 5	
5	Inputs, module 2		Inputs, module 6	
6	Inputs, module 3		Inputs, module 7	
7	Inputs, module 4		Inputs, module 8	
8	Inputs, module 5		Inputs, module 9	
9	Inputs, module 6		Inputs, module 10	
10	Inputs, module 7		Inputs, module 11	
11	Inputs, module 8		Inputs, module 12	
12	Inputs, module 9		Outputs, module 1	2 (12 bytes)
13	Inputs, module 10		Outputs, module 2	
14	Inputs, module 11		Outputs, module 3	
15	Inputs, module 12		Outputs, module 4	
16	Outputs, module 1		Outputs, module 5	
17	Outputs, module 2		Outputs, module 6	
18	Outputs, module 3		Outputs, module 7	
19	Outputs, module 4		Outputs, module 8	
20	Outputs, module 5		Outputs, module 9	
21	Outputs, module 6		Outputs, module 10	
22	Outputs, module 7		Outputs, module 11	
23	Outputs, module 8		Outputs, module 12	
24	Outputs, module 9		Logic result 0	3 (12 bytes)
25	Outputs, module 10		Logic result 1	
26	Outputs, module 11		Logic result 2	
27	Outputs, module 12		Logic result 3	
28	Direct gateway output values 1		Direct gateway output values 1	
29	Direct gateway output values 2		Direct gateway output values 2	
30	Direct gateway output values 3		Direct gateway output values 3	
31	Direct gateway output values 4		Direct gateway output values 4	
32 ... 35	Not assigned		Not assigned	
36 ... 47	Not assigned		Not assigned	4 (12 bytes)
48 ... 49	Not assigned		Not assigned	5 (2 bytes)

Corresponding information for the FX0-GETC: see table 43, page 61 and for the FX0-GCAN: see table 82, page 103.

The assignment of the data sets and assemblies can be configured for all gateways as shown in the following section.

5.5 Process image configuration

This section shows how to configure the process image, which the Flexi Soft gateway transfers to the network. For more information, see the operating instructions titled “Flexi Soft in the Flexi Soft Designer Configuration Software” (SICK part number 8012998).

Configuration of the Flexi Soft gateway for data routing is shown in the dialog box for gateway configuration.

- ► Click on the **Interfaces** button above the main window and select the required gateway or double-click on the required gateway in the **Hardware configuration** to open the dialog box for gateway configuration.
- ► Click on the **Flexi Soft to network** button to display the routing configuration.

The default setting is as follows (example for Modbus TCP):

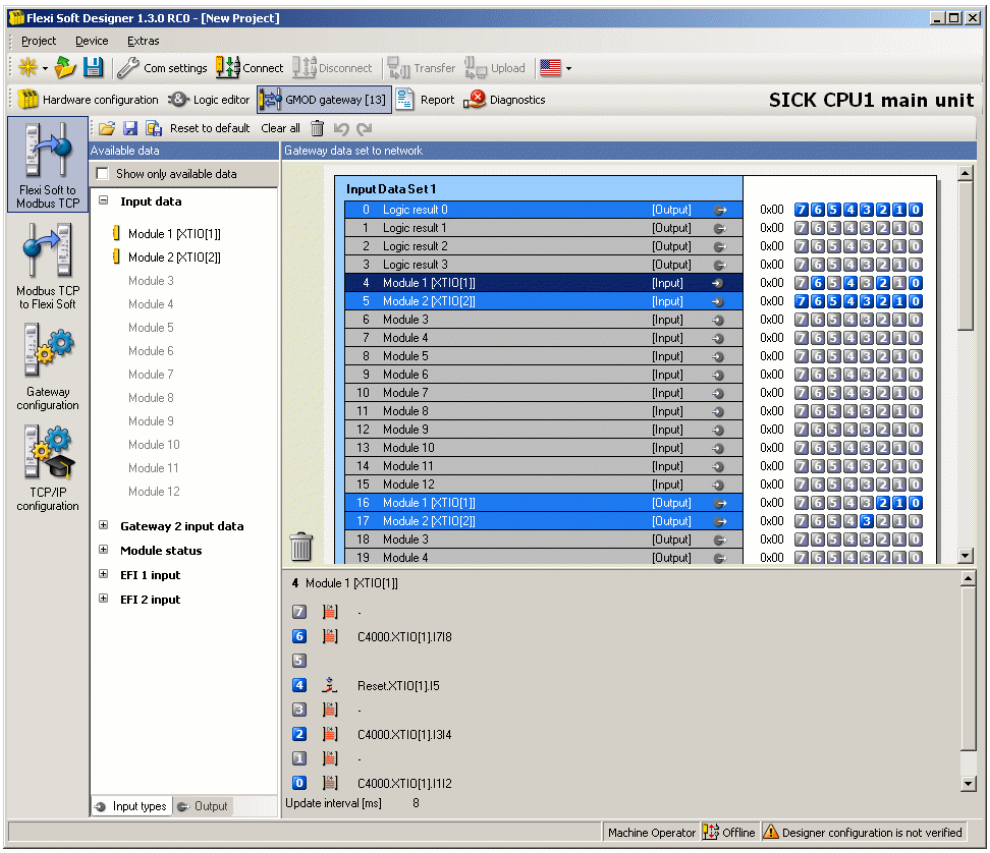

Figure 38: Default configuration for the operating data transmitted to the network

This dialog box is divided into the following areas: the **Available data** area on the left and the **Gateway data set to network** area on the right. This area shows the available bytes at the top and the **Tag names** for the selected byte at the bottom. The toolbar is located above the dialog box.

5.5.1 The toolbar

Figure 39: Toolbar for the routing configuration

The toolbar contains buttons for the following actions (working from left to right):

- The **Load user-specific configuration** and **Save user-specific configuration** buttons can be used to load/save the routing configuration in XML format. When a routing configuration is loaded, any unsaved changes to the routing configuration will be lost. This command cannot be undone.
- With the **Import** and **Export** buttons, the tag names used can be imported/exported as a CSV file or in a network-specific format, such as SIEMENS SEQ files (*.seq) for PROFIBUS or PROFINET IO. This allows the tag names to be imported into a PLC program and used there as well.

NOTE
The **Import** button is only available when the routing is configured for the network to Flexi Soft direction.

- **Reset to default** restores the default routing configuration. Confirm the command with **Yes**. In this case, any unsaved changes to the routing configuration will be lost. This command cannot be undone.
- **Clear all** deletes the routing configuration, i.e., all bytes assigned in the **Gateway data set to network** area are cleared.
- **Remove routing** deletes the byte that is currently selected in the **Gateway data set to network** area.
- The **Undo** and **Redo** buttons enable the most recent changes made to the routing configuration to be undone or redone.

5.5.2 Available Data area

This area contains all the sources from which data can be routed into the network. It is divided into two views, which contain the available input and output data. The tab at the bottom can be used to switch between these two views.

- The **Input types** view contains the input values of the connected Flexi Soft modules and EFI-enabled devices, plus the module status data. If the Flexi Soft system contains a second gateway, the input data for this gateway (i.e., the data received by the second gateway from the network) can also be found here.
- The **Output** view contains the output values of the connected Flexi Soft modules and EFI-enabled devices, plus the **Logic results** from the logic editor.

All sources supported by the current configuration are shown in black:

- Connected Flexi Soft modules
- Connected EFI-enabled devices
- Configured logic results [1)]
- Gateway input data that is made available in the system by another gateway

Sources that are not supported by the current configuration are shown in gray. You can use the **Show only available data** box at the top left to hide sources which are not used.

Sources that provide "live" data are identified with a symbol on the left next to the text (see figure 38, page 140).

5.5.3 Gateway data set to network area

This area contains the routing table. It displays the current content of the input data modules for the Flexi Soft gateway. Bytes and bits that are highlighted in blue contain “live” system data if the source is supported by the hardware configuration. Bytes

1) In the default configuration, only the first byte of the logic results (logic result 0) is active and available. More output bits for logic results can be activated in the logic editor if required (see the operating instructions titled “Flexi Soft in the Flexi Soft Designer Configuration Software” (SICK part number 8012998)).

shown in gray are assigned by default but do not have any data assigned to them because the current hardware configuration does not support the sources. Highlighted bytes are not assigned.

Adding a data byte to the routing table

► Drag an element (e.g., a 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, it must be cleared first by deleting the byte that is currently assigned to it, or by moving this byte to another position in the table.

NOTE
The same byte can be used multiple times within the routing table.

Deleting a data byte from the routing table

► Drag the byte to be deleted to the trash symbol at the bottom left corner of the **Gateway data set to network** area.

Or:

► Select the byte to be deleted by clicking on it. Then click on the **Remove routing** button on the toolbar.

Or:

► In the context menu of the byte to be deleted, select the **Remove routing** command.

Moving a data byte to another position in the routing table

► Drag the byte to be moved to the desired position. If the desired position is not free, it must be cleared first by deleting the byte that is currently assigned to it, or by moving this byte to another position in the table.

5.5.4 The Tag names area

This area displays the tag names of all the bits of the byte currently selected in the **Available data** or **Gateway data set to network** area. You can enter and edit the tag names in either the logic editor or the tag name editor.

Only the tag names of the direct gateway output values can be edited in the **Tag names** area of the configuration window for the Flexi Soft to network routing direction.

5.5.5 Direct gateway output values

Values (e.g., logic results) can be written directly from the logic editor to a gateway. In the default setting for the process image, four bytes are reserved for these direct gateway output values, which can be found on the **Outputs** tab in the logic editor.

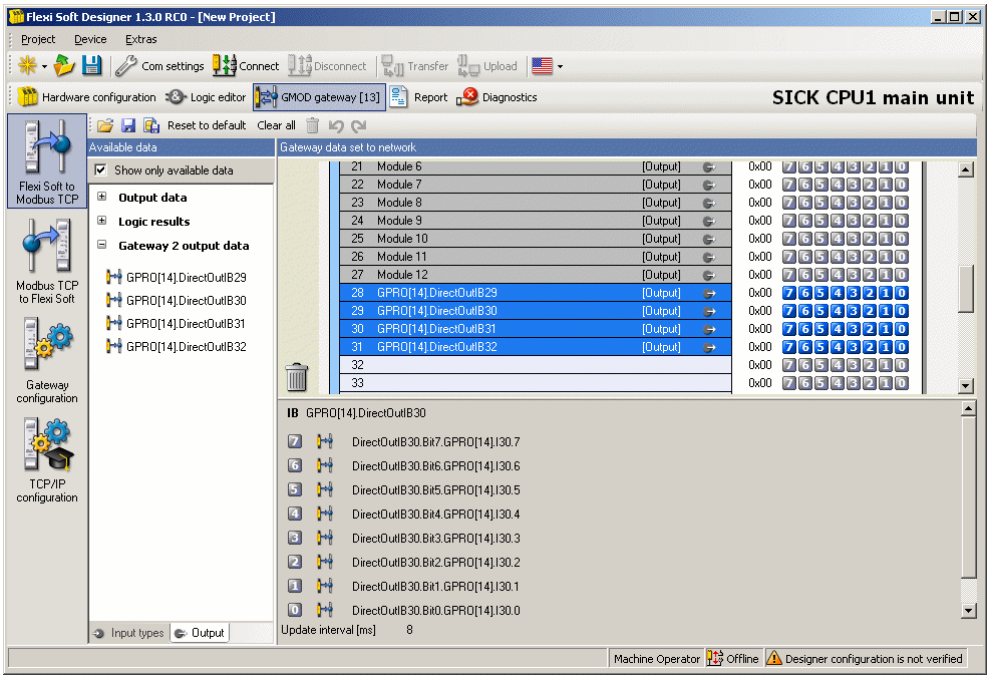

Figure 40: Direct gateway output values in the process image (default)

It is possible to configure every byte as a direct gateway output value. For this purpose, tag names must be assigned to the bits used.

Configuring direct gateway output values

- ► Click on a free byte in the **Gateway data set to network** area to select it. Free bytes are highlighted.
- ► Check the **Use direct editing** box in the top left corner of the **Tag names** area. The tag names for the selected byte can now be edited.
- ► Enter a tag name for the selected byte.
- ► Enter tag names for the individual bits of the selected byte.

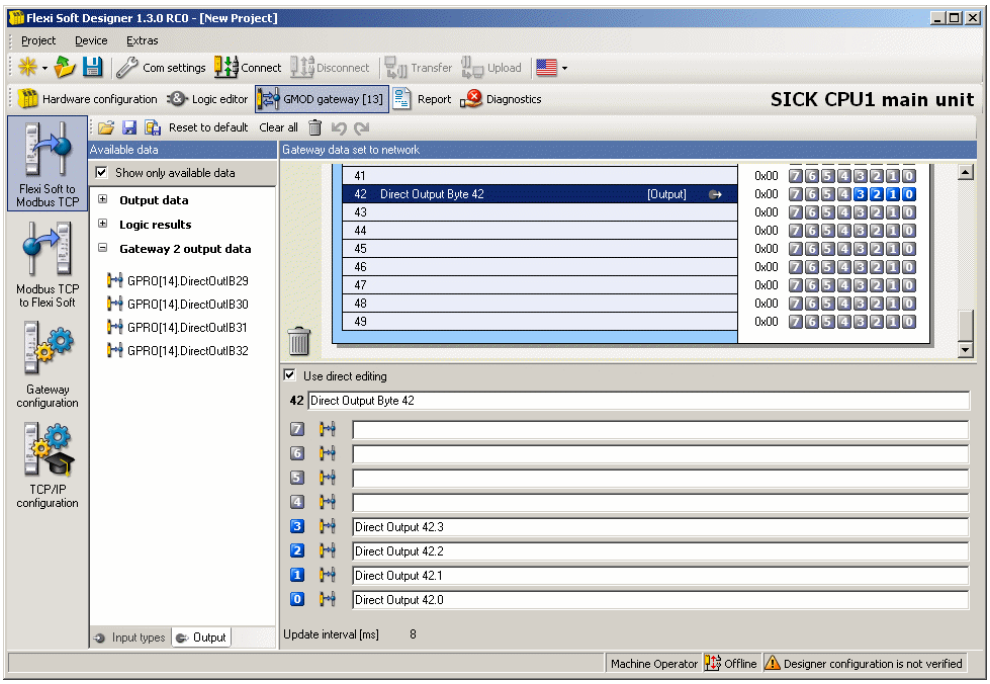

Figure 41: Configuring direct gateway output values in the process image

All bits with a tag name are available on the **Outputs** tab in the logic editor.

NOTE
Predefined direct gateway output values can be edited in the same way.

5.5.6 Configuring the output data (network to Flexi Soft)

Activating incoming data bits

► In the menu on the left, click on the button for the network to Flexi Soft transmission direction. The following dialog box is displayed:

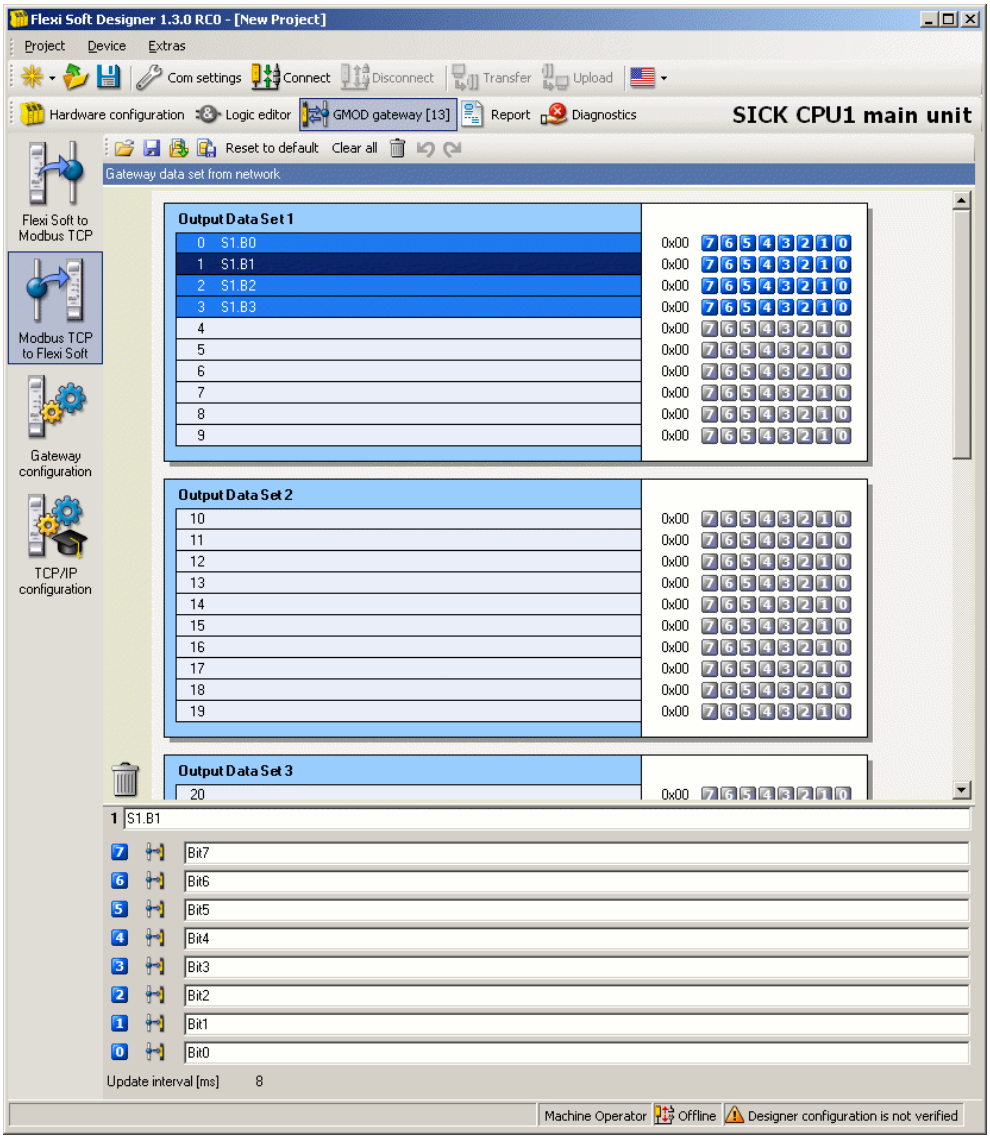

Figure 42: Modbus TCP to Flexi Soft dialog box of the FX0-GMOD

The **Gateway data set from network** area is shown in this dialog box. This area shows the current configuration of the output data sets at the top and the **Tag names** for the selected byte at the bottom.

► Select a byte in the **Gateway data set from network** area.
► Enter a **Tag name** for each bit of the selected byte that is to be used.

All bits with a tag name are available in the logic editor or in the process image of a second gateway.

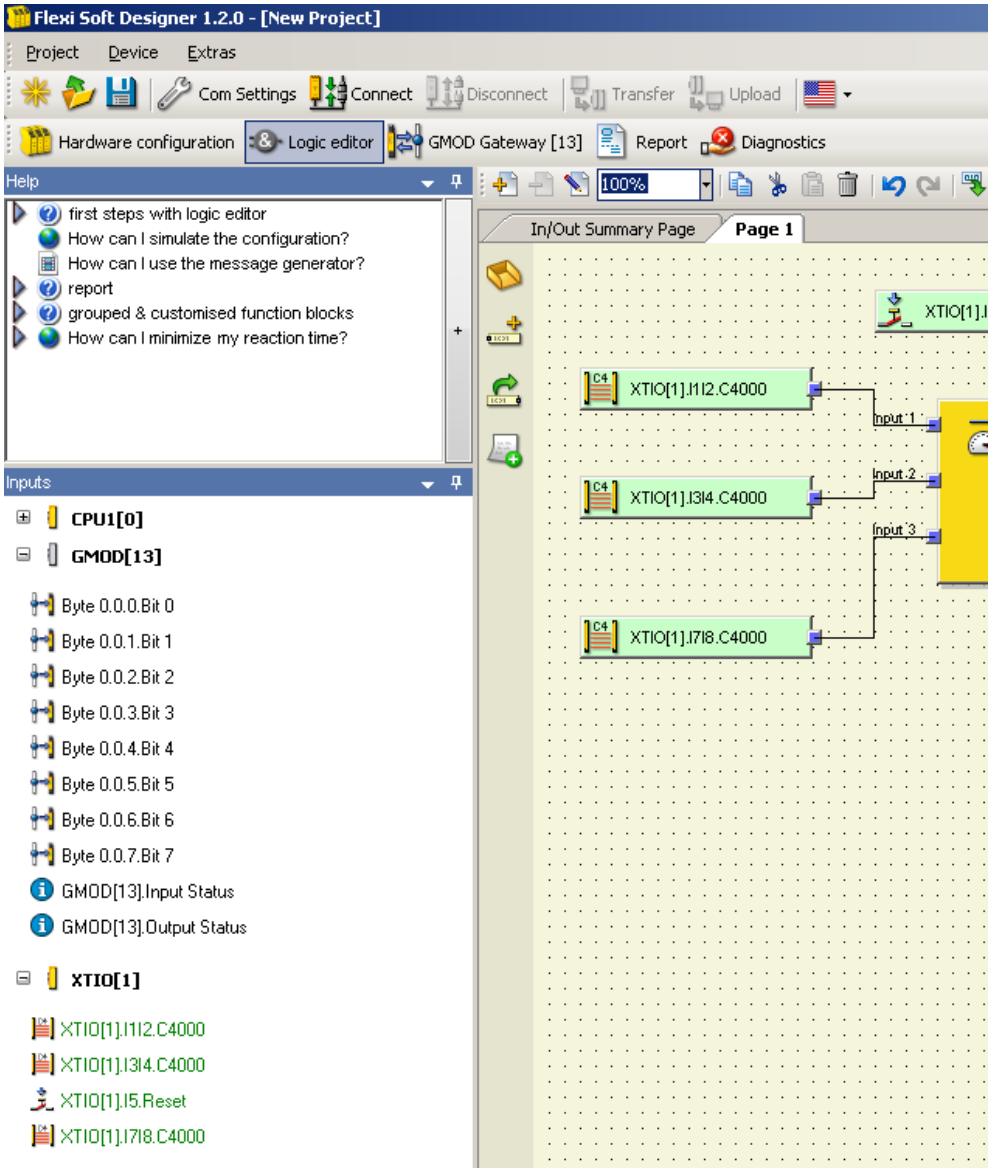

Figure 43: Tag names of incoming bits in the logic editor of the main module

5.5.7 Loading and saving a configuration

The **Load user-specific configuration** and **Save user configuration** buttons can be used to load/save the routing configuration in XML format. When a routing configuration is loaded, any unsaved changes to the routing configuration will be lost. This command cannot be undone.

5.5.8 Importing and exporting a configuration

With the **Import** and **Export** buttons, you can import or export a configuration, including the tag names used, as a CSV file or in a network-specific format, such as SIEMENS SEQ files (*.seq) for PROFIBUS or PROFINET IO. This allows the tag names used in a Flexi Soft project to be imported into a PLC program and used there as well.

When a configuration is imported, this will overwrite the existing configuration. This command cannot be undone.

NOTE
The **Import** command is only available when the routing is configured for the network to Flexi Soft direction.

5.6 Online monitoring of the operating data

If the Flexi Soft system is online and in operation, then the operating data is displayed online in the window for the gateway configuration.

- ► Click on the **Interfaces** button above the main window and select the required gateway or double-click on the required gateway in the **Hardware configuration** to open the dialog box for gateway configuration.
- ► Click on the button for the **Flexi Soft to network** direction or the button for the **Network to Flexi Soft** direction to display the routing table for the input or output data to be monitored.

For both directions – **Flexi Soft to network** and **Network to Flexi Soft** – inactive bits are shown in gray and active bits in green:

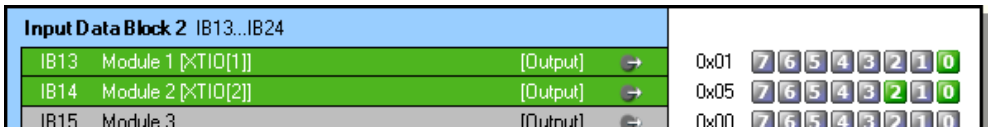

Figure 44: Active and inactive bits in the online process image

In the view for the **Flexi Soft to network** direction, bits that are deactivated due to an error are displayed in red. This can occur at the outputs of an FX3-XTIO module, for example, if the voltage supply of this module is defective:

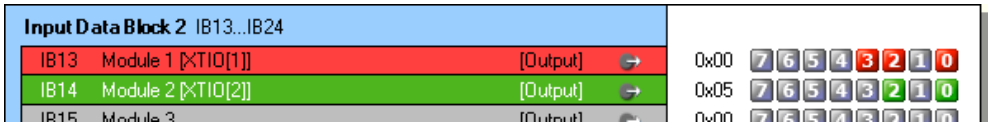

Figure 45: Inactive network input bits as the result of an error

In the view for the **Network to Flexi Soft** direction, bits, which have not been assigned a tag name (so that they cannot be edited in the logic editor) but which are included in the process image that receives the Flexi Soft gateway from the PLC, are highlighted in yellow:

Figure 46: Network output bits without assigned tag name in the online process image

NOTE
Force mode

The Flexi Soft gateways always display the actual physical status of the inputs and outputs of the connected modules and devices. This means that, even when force mode is active and inputs which are physically **Low** are forced to **High** (or vice versa), the actual physical status of the inputs is transferred to the PLC and not the (virtual) forced status. However, if one or several outputs change their status as a result of the forcing of one or several inputs, then the changed status of these outputs is also transferred to the PLC as the actual physical status of the device outputs has changed. For detailed information on force mode, please refer to the operating instructions titled “Flexi Soft in the Flexi Soft Designer Configuration Software” (SICK part number 8012998).

6 Troubleshooting

Detailed information on diagnostics of the Flexi-Soft system is contained in the operating instructions "Flexi Soft in the Flexi Soft Designer configuration software" (SICK part number 8012998).

6.1 The FX0-GENT EtherNet/IP™ gateway

Table 145: Troubleshooting the FX0-GENT

Fault	Possible cause	Possible measures
The computer is unable to establish a connection to the Flexi Soft gateway.	The supply voltage of the FX0-GENT is too low or missing. The FX0-GENT is not located on the same physical network as the computer. A different subnet mask has been configured in the TCP/IP settings of the computer. The FX0-GENT has been configured already at some point and either has a fixed IP address or an IP address that has been assigned by an unrecognized DHCP server.	Switch on the voltage supply. Check the Ethernet cabling and network settings of the computer and make any necessary corrections. Set the subnet mask of the computer to 255.255.0.0 (as-delivered state of the FX0-GENT). Check the communication settings of the FX0-GENT.
The FX0-GENT is not supplying any data. PWR ● Green LINK/ACT ●/☼ Green STATUS [1] ☼ Red/green	The FX0-GENT has been configured for transmitting data to the PLC but no Ethernet communication has been established yet or this communication is faulty. Duplicate IP address detected. Another device on the network has the same IP address.	At least one Ethernet connection must be established. Set up an Ethernet connection on the computer and check the Ethernet cabling. Check the Ethernet settings for the Flexi Soft system on the PLC and in the configuration software. If no Ethernet communication is required, deactivate the Ethernet connections/PLC interfaces on the FX0-GENT. Correct the IP address. Then switch the device off and back on again.
The FX0-GENT is not supplying any data. PWR ● Green LINK/ACT ●/☼ Green STATUS [1] ☼ Red (1 Hz)	Configuration required. The configuration has not yet been transferred in full.	Configure the FX0-GENT and transfer the configuration to the device. Wait until the configuration has been fully transferred.
The FX0-GENT is not supplying any data. PWR ● Green LINK/ACT ●/☼ Green STATUS [1] ● Green	No data set has been activated. No Ethernet communication interface has been activated.	Activate at least one data set.
The FX0-GENT is not supplying any data. PWR ● Green LINK/ACT ●/☼ Green STATUS [1] ☼ Green (1 Hz)	The FX0-GENT is in the “Stopped” state.	Start the main module (switch to the “Run” state).

Fault	Possible cause	Possible measures
The FX0-GENT was functioning correctly following configuration but has suddenly stopped supplying data. PWR ● Green LINK/ACT ●/☼ Green STATUS [1] ☼ Red/ green	The FX0-GENT is being operated in slave mode, with the IP address assigned by a DHCP server. Following a restart of the FX0-GENT or the DHCP server, a different IP address was assigned to the FX0-GENT that is not recognized by the PLC.	Either assign a fixed IP address to the FX0-GENT or allocate a fixed IP address for the FX0-GENT on the DHCP server (assigned manually via the MAC address of the FX0-GENT).
The FX0-GENT is in the “Serious error” state. PWR ● Green LINK/ACT ●/☼ Green STATUS [1] ☼ Red (2 Hz)	Internal device error on the FX0-GENT. The FX0-GENT is in an incorrect position. No Flexi Soft gateways are supported by the firmware version of the main module.	Switch the voltage supply for the Flexi Soft system off and then back on again. Check whether the FX0-GPNT is positioned correctly in the Flexi Soft system. Use the configuration software to check the diagnostic messages. Use a main module with the required firmware version. If the fault persists, replace the gateway.
The FX0-GENT/ Flexi Soft system is in the “Serious error” state. PWR ● Green LINK/ACT ☼ Green STATUS [1] ● Red	The FX0-GENT has not been connected correctly to the other Flexi Soft modules. The module connector is contaminated or damaged. The FX0-GENT is in an incorrect position. There is an internal serious error on another Flexi Soft module.	Plug in the FX0-GENT correctly. Check whether the FX0-GPNT is positioned correctly in the Flexi Soft system. Clean the male and female connectors. Switch the voltage supply back on. Check the other Flexi Soft modules.

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

6.2 The FX0-GMOD Modbus TCP gateway

Table 146: Troubleshooting the FX0-GMOD

Fault	Possible cause	Possible measures
The computer is unable to establish a connection to the Flexi Soft gateway.	The supply voltage of the FX0-GMOD is too low or missing. The FX0-GMOD is not located on the same physical network as the computer. A different subnet mask has been configured in the TCP/IP settings of the computer. The FX0-GMOD has been configured already at some point and either has a fixed IP address or an IP address that has been assigned by an unrecognized DHCP server.	Switch on the voltage supply. Check the Ethernet cabling and network settings of the computer and make any necessary corrections. Set the subnet mask of the computer to 255.255.0.0 (as-delivered state of the FX0-GMOD). Check the communication settings of the FX0-GMOD.

Fault	Possible cause	Possible measures
The FX0-GMOD is not supplying any data. PWR ● green LINK/ACT ●/☼ green STATUS [1] ☼ red/green	The FX0-GMOD has been configured for transmitting data to the PLC but no Ethernet communication has been established yet or this communication is faulty. Duplicate IP address detected. Another device on the network has the same IP address.	At least one Ethernet connection must be established. Set up an Ethernet connection on the computer and check the Ethernet cabling. Check the Ethernet settings for the Flexi Soft system on the PLC and in the configuration software. If no Ethernet communication is required, deactivate the Ethernet connections/PLC interfaces on the FX0-GMOD. Correct the IP address. Then switch the device off and back on again. On the PLC, check the Modbus port number for Modbus TCP communication. The Modbus port number must be set to 502. (Do not confuse this with the TCP/IP socket port number, which must be set to a value > 1023.)
The FX0-GMOD is not supplying any data. PWR ● green LINK/ACT ●/☼ green STATUS [1] ☼ red (1 Hz)	Configuration required. The configuration has not yet been transferred in full.	Configure the FX0-GMOD and transfer the configuration to the device. Wait until the configuration has been fully transferred.
The FX0-GMOD is not supplying any data. PWR ● green LINK/ACT ●/☼ green STATUS [1] ● green	No data set has been activated. No Ethernet communication interface has been activated.	Activate at least one data set.
The FX0-GMOD is not supplying any data. PWR ● green LINK/ACT ●/☼ green STATUS [1] ☼ green (1 Hz)	The FX0-GMOD is in the “Stopped” state.	Start the main module (switch to the “Run” state).
The FX0-GMOD was functioning correctly following configuration but has suddenly stopped supplying data. PWR ● green LINK/ACT ●/☼ green STATUS [1] ☼ red/green	The FX0-GMOD is being operated in slave mode, with the IP address assigned by a DHCP server. Following a restart of the FX0-GMOD or the DHCP server, a different IP address was assigned to the FX0-GMOD that is not recognized by the PLC.	Either assign a fixed IP address to the FX0-GMOD or allocate a fixed IP address for the FX0-GMOD on the DHCP server (assigned manually via the MAC address of the FX0-GMOD).
The FX0-GMOD is in the “Serious error” state. PWR ● green LINK/ACT ●/☼ green STATUS [1] ☼ red (2 Hz)	Internal device error on the FX0-GMOD. No Flexi Soft gateways are supported by the firmware version of the main module.	Switch the voltage supply for the Flexi Soft system off and then back on again. Use the configuration software to check the diagnostic messages. Use a main module with the required firmware version. If the fault persists, replace the gateway.

Fault	Possible cause	Possible measures
The FX0-GMOD/ Flexi Soft system is in the “Serious error” state. PWR ● green LINK/ACT ☼ green STATUS [1] ● red	The FX0-GMOD has not been connected correctly to the other Flexi Soft modules. The module connector is contaminated or damaged. There is an internal serious error on another Flexi Soft module.	Plug in the FX0-GMOD correctly. Clean the male and female connectors. Switch the voltage supply back on. Check the other Flexi Soft modules.

1) For older device versions, the LED is called MS.

6.3 The FX0-GPNT PROFINET IO gateway

Table 147: Troubleshooting the FX0-GPNT

Fault	Possible cause	Possible measures
The computer is unable to establish a connection to the Flexi Soft gateway.	The supply voltage of the FX0-GPNT is too low or missing. The FX0-GPNT is not located on the same physical network as the computer. A different subnet mask has been configured in the TCP/IP settings of the computer. The FX0-GPNT has been configured already at some point and either has a fixed IP address or an IP address that has been assigned by an unrecognized DHCP server. [1]	Switch on the voltage supply. Check the Ethernet cabling and network settings of the computer and make any necessary corrections. Set the subnet mask of the computer to 255.255.0.0 (as-delivered state of the FX0-GPNT). Check the communication settings of the FX0-GPNT. Check the IP address of the FX0-GPNT.
The FX0-GPNT is not supplying any data. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Red/ green [1]	The FX0-GPNT has been configured for transmitting data to the PLC but no Ethernet communication has been established yet or this communication is faulty. Duplicate IP address detected. Another device on the network has the same IP address. Incorrectly formatted PROFINET IO device name.	At least one Ethernet connection must be established. Set up an Ethernet connection on the computer and check the Ethernet cabling. Check the Ethernet settings for the Flexi Soft system on the PLC and in the configuration software. If no Ethernet communication is required, deactivate the Ethernet connections/PLC interfaces on the FX0-GPNT. Correct the IP address. Then switch the device off and back on again. Compare the device name on the PROFINET IO master with the one on the FX0-GPNT.
The FX0-GPNT is not supplying any data. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Red (1 Hz)	Configuration required. The configuration has not yet been transferred in full.	Configure the FX0-GPNT and transfer the configuration to the device. Wait until the configuration has been fully transferred.
The FX0-GPNT is not supplying any data. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Green (1 Hz)	No data set has been activated. The Flexi Soft system is in the “Stopped” state.	Activate at least one data set. Start the main module (switch to the “Run” state).

Fault	Possible cause	Possible measures
The FX0-GPNT is not supplying any data. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Green (2 Hz)	LED flashing at the request of the PROFINET IO master for the purpose of physically identifying the device.	Stop the LED from flashing with the SIEMENS SIMATIC Manager software or switch the voltage supply for the Flexi Soft system off and then back on again.
The FX0-GPNT was functioning correctly following configuration but has suddenly stopped supplying data. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Red/green [1]	The FX0-GPNT is being operated in slave mode, with the IP address assigned by a DHCP server. Following a restart of the FX0-GPNT or the DHCP server, a different IP address was assigned to the FX0-GPNT that is not recognized by the PLC. [1]	Either assign a fixed IP address to the FX0-GPNT or allocate a fixed IP address for the FX0-GPNT on the DHCP server (assigned manually via the MAC address of the FX0-GPNT).
The FX0-GPNT is in the “Serious error” state. PWR ● Green LINK/ACT ●/☼ Green STATUS ☼ Red (2 Hz)	Internal device error on the FX0-GPNT. The FX0-GPNT is in an incorrect position. No Flexi Soft gateways are supported by the firmware version of the main module.	Switch the voltage supply for the Flexi Soft system off and then back on again. Check whether the FX0-GPNT is positioned correctly in the Flexi Soft system. Use the configuration software to check the diagnostic messages. Use a main module with the required firmware version. If the fault persists, replace the gateway.
The FX0-GPNT/ Flexi Soft system is in the “Serious error” state. PWR ● Green LINK/ACT ☼ Green STATUS ● Red	The FX0-GPNT has not been connected correctly to the other Flexi Soft modules. The module connector is dirty or damaged. The FX0-GPNT is in an incorrect position. There is an internal serious error on another Flexi Soft module.	Plug in the FX0-GPNT correctly. Check whether the FX0-GPNT is positioned correctly in the Flexi-Soft system. Clean the male and female connectors. Switch the voltage supply back on. Check the other Flexi Soft modules.

1) FX0-GPNT with firmware ≥ V3.00.0 do not support DHCP.

2) If the firmware version is ≥ V2.00.0, you can use the configuration software to stop the STATUS LED from flashing red/green ☼. In this case, the STATUS LED will light up steady ● green if the configuration is valid.

6.4 The FX0-GETC EtherCAT gateway

Table 148: Troubleshooting the FX0-GETC

Fault	Possible cause	Possible measures
The computer is unable to establish a connection to the Flexi Soft gateway.	The supply voltage of the FX0-GETC is too low or missing.	Switch on the voltage supply. Check the communication settings of the FX0-GETC.
The FX0-GETC is not supplying any data. MS ☼ Red (1 Hz) ERR ☼ Red (2.5 Hz) RUN ○ OFF	Configuration required. The configuration has not yet been transferred in full.	Configure the FX0-GETC and transfer the configuration to the device. Wait until the configuration has been fully transferred.

Fault	Possible cause	Possible measures
The FX0-GETC is not supplying any data. MS ● Red/green ERR ○ OFF RUN ☼ Green (2.5 Hz)	No input PDO activated.	Activate an input PDO.
The FX0-GETC is not supplying any data. MS ☼ Green (1 Hz) ERR ○ OFF RUN ● Green	The Flexi Soft system is in the “Stopped” state.	Start the main module (switch to the “Run” state).
The FX0-GETC is not supplying any data. MS ● Green ERR ○ OFF RUN ● Green	The EtherCAT PLC is in the “Stopped” state.	Start the EtherCAT PLC (switch to the “Run” state).
The FX0-GETC is in the “Serious error” state. MS ☼ Red (2 Hz) ERR ● Red RUN ○ OFF	Internal device error on the FX0-GETC. No Flexi Soft gateways are supported by the firmware version of the main module.	Switch the voltage supply for the Flexi Soft system off and then back on again. Use the configuration software to check the diagnostic messages. Use a main module with the required firmware version. If the fault persists, replace the gateway.
The FX0-GETC/ Flexi Soft system is in the “Serious error” state. MS ● Red ERR ● Red RUN ○ OFF	The FX0-GETC has not been connected correctly to the other Flexi Soft modules. The module connector is contaminated or damaged. There is an internal serious error on another Flexi Soft module.	Plug in the FX0-GETC correctly. Clean the male and female connectors. Switch the voltage supply back on. Check the other Flexi Soft modules. Use the configuration software to check the diagnostic messages.

6.5 The FX0-GPRO PROFIBUS DP gateway

Table 149: Troubleshooting the FX0-GPRO

Fault	Possible cause	Possible measures
The computer is unable to establish a connection to the Flexi Soft gateway.	The supply voltage of the FX0-GPRO is too low or missing.	Switch on the voltage supply. Check the communication settings of the FX0-GPRO.
The FX0-GPRO is not supplying any data. PWR ● Green BF ○ OFF MS ☼ Red (1 Hz)	Configuration required. The configuration has not yet been transferred in full.	Configure the FX0-GPRO and transfer the configuration to the device. Wait until the configuration has been fully transferred.
The FX0-GPRO is not supplying any data. PWR ● Green BF ○ OFF MS ● Green	No data set has been activated.	Activate at least one data set.

Fault	Possible cause	Possible measures
The FX0-GPRO is not supplying any data. PWR ● Green BF ○ OFF/● Red MS ☼ Green (1 Hz)	The FX0-GPRO is in the “Stopped” state.	Start the main module (switch to the “Run” state).
The FX0-GPRO is not supplying any data. PWR ● Green BF ○ OFF MS ● Green	The PROFIBUS master is in the “Stopped” state.	Start the PROFIBUS master (switch to the “Run” state).
The FX0-GPRO was functioning correctly following configuration but has suddenly stopped supplying data. PWR ● Green BF ● Red MS ☼ Red/green	The PROFIBUS hardware address of the FX0-GPRO has been changed. The PROFIBUS cable has been interrupted.	Check the PROFIBUS address settings on the hardware. Check the PROFIBUS cable. Check the PROFIBUS master.
The FX0-GPRO is in the “Serious error” state. PWR ● Green BF ● Red MS ☼ Red (2 Hz)	Internal device error on the FX0-GPRO. No Flexi Soft gateways are supported by the firmware version of the main module.	Switch the voltage supply for the Flexi Soft system off and then back on again. Use the configuration software to check the diagnostic messages. Use a main module with the required firmware version. If the fault persists, replace the gateway.
The FX0-GPRO/ Flexi Soft system is in the “Serious error” state. PWR ● Red BF ○ OFF MS ● Red	The FX0-GPRO has not been connected correctly to the other Flexi Soft modules. The module connector is contaminated or damaged. There is an internal serious error on another Flexi Soft module.	Plug in the FX0-GPRO correctly. Clean the male and female connectors. Switch the voltage supply back on. Check the other Flexi Soft modules.

6.6 The FX0-GCAN CANopen gateway

Table 150: Troubleshooting the FX0-GCAN

Fault	Possible cause	Possible measures
The computer is unable to establish a connection to the Flexi Soft gateway.	The supply voltage of the FX0-GCAN is too low or missing.	Switch on the voltage supply. Check the communication settings of the FX0-GCAN.
The FX0-GCAN is not supplying any data. PWR ● Green NS ○ OFF MS ☼ Red (1 Hz)	Configuration required, the node guarding or heartbeart messages have not been sent. The configuration has not yet been transferred in full.	Configure the FX0-GCAN and transfer the configuration to the device. Wait until the configuration has been fully transferred.
The FX0-GCAN is not supplying any data. PWR ● Green NS ● Green MS ☼ Red (1 Hz)	The configuration has not yet been transferred in full.	Wait until the configuration has been fully transferred.

Fault	Possible cause	Possible measures
The FX0-GCAN is not supplying any data. PWR ● Green NS ● Green MS ☼ Red/green	No PDO transmission has taken place since switching on	Start PDO transmission. Transmit the PDO via SDO 6000h or SDO 6200h.
The FX0-GCAN is not supplying any data. PWR ● Green NS ☼ Green MS ☼ Red/green	No PDO transmission has taken place since switching on Incorrect data transmission rate (CAN transceiver may be in the “Passive error” state) Wrong node ID or CANopen address The CAN cable has been interrupted.	Start PDO transmission. Transmit the PDO via SDO 6000h or SDO 6200h. Check and correct the data transmission rate. Check and correct the CANopen address. Check the CANopen cabling.
The FX0-GCAN is not supplying any data. PWR ● Green NS ○ OFF/● Red/● Green MS ☼ Green (1 Hz)	The FX0-GCAN is in the “Idle” state. The node guarding or heartbeart messages have not been sent. The Flexi Soft configuration has not been verified and the main module has been stopped.	Start the main module (switch to the “Run” state). Use the configuration software to verify the configuration and start the main module.
The FX0-GCAN is not supplying any data. PWR ● Green NS ● Green MS ○ OFF	Supply voltage too low	Check the supply voltage.
The FX0-GCAN is not supplying any data. PWR ● Red NS ● Red MS ● Red	Brief supply voltage drop	Check the supply voltage. Reset the Flexi Soft system.
The FX0-GCAN is not supplying any data. PWR ● Green NS ☼ Green (1 Hz) MS ☼ Green (1 Hz)	Wrong node ID or CANopen address Incorrect data transmission rate (CAN transceiver may be in the “Passive error” state), FX0-GCAN is in the “Idle” state.	Check and correct the CANopen address. Check and correct the data transmission rate.
The FX0-GCAN is not supplying any data. PWR ● Green NS ● Red MS ☼ Red/green	The data transmission rate is incorrect and the FX0-GCAN transceiver is in the “Busoff” state (hardware problem on the physical CAN layer). The CAN cable has been interrupted.	Check and correct the data transmission rate. Check the CANopen cabling. Reset the Flexi Soft system.
The FX0-GCAN is not supplying any data. PWR ● Green NS ☼ Green (1 Hz) MS ● Green	The CANopen master is in the “Stop” or “Pre-operational” status. Unable to initialize another slave during bus system initialization. The CANopen status of the FX0-GCAN is “Pre-operational”. Wrong node ID or CANopen address.	Start the CANopen master (switch to the “Operational” CANopen status). Check that all the slaves on the bus are switched on. Check the CANopen cabling. Check whether the CAN master starts automatically. Check and correct the CANopen address.

Fault	Possible cause	Possible measures
The FX0-GCAN is not supplying any data. PWR ● Green NS ● Red MS ● Green	The FX0-GCAN transceiver is in the “Passive error” state. The CAN cable has been interrupted.	Check the CANopen cabling. Use the configuration software to check the diagnostic messages. Reset the Flexi Soft system.
The FX0-GCAN is not supplying any data. PWR ● Green NS ☼ Red (1 Hz) MS ☼ Red/green	Node guarding or heartbeart consumer failure The guarding configuration has been changed.	Check the CANopen cabling. Check the life guarding time (life time factor ≥ 1). Check the heartbeat consumer time (should be ≥ 1.5 × heartbeat producer time). Use the configuration software to check the diagnostic messages. Reset the Flexi Soft system.
The FX0-GCAN is in the “Serious error” state. PWR ● Green NS ● Red MS ☼ Red (2 Hz)	Internal device error on the FX0-GCAN. No Flexi Soft gateways are supported by the firmware version of the main module.	Switch the voltage supply for the Flexi Soft system off and then back on again. Use the configuration software to check the diagnostic messages. Use a main module with the required firmware version. If the fault persists, replace the gateway.
The FX0-GCAN/ Flexi Soft system is in the “Serious error” state. PWR ● Red NS ○ OFF MS ● Red	The FX0-GCAN has not been connected correctly to the other Flexi Soft modules. The module connector is contaminated or damaged. There is an internal serious error on another Flexi Soft module.	Plug in the FX0-GCAN correctly. Clean the male and female connectors. Switch the voltage supply back on. Check the other Flexi Soft modules.

6.7 The FX0-GDEV DeviceNet gateway

Table 151: Troubleshooting the FX0-GDEV

Fault	Possible cause	Possible measures
The computer is unable to establish a connection to the Flexi Soft gateway.	The supply voltage of the FX0-GDEV is too low or missing.	Switch on the voltage supply. Check the communication settings of the FX0-GDEV.
The FX0-GDEV is not supplying any data. PWR ● Green NS ○ OFF MS ☼ Red (1 Hz)	Configuration required, the node guarding or heartbeart messages have not been sent. The configuration has not yet been transferred in full.	Configure the FX0-GDEV and transfer the configuration to the device. Wait until the configuration has been fully transferred.
The FX0-GDEV is not supplying any data. PWR ● Green NS ● Green MS ☼ Red (1 Hz)	The configuration has not yet been transferred in full.	Wait until the configuration has been fully transferred.
The FX0-GDEV is not supplying any data. PWR ● Green NS ● Green MS ☼ Red/green	No data transmission has taken place since switching on	Start data transmission.

Fault	Possible cause	Possible measures
The FX0-GDEV is not supplying any data. PWR ● Green NS ☼ Green MS ☼ Red/green	No data transmission has taken place since switching on Incorrect data transmission rate Wrong node ID or DeviceNet address The cable has been interrupted.	Start data transmission. Check and correct the data transmission rate. Check and correct the node ID and DeviceNet address. Check the cabling.
The FX0-GDEV is not supplying any PDO data. PWR ● Green NS ○ OFF/● Red/● Green MS ☼ Green (1 Hz)	The FX0-GDEV is in the “Idle” state. The node guarding or heartbeart messages have not been sent. The Flexi Soft configuration has not been verified and the main module has been stopped.	Start the main module (switch to the “Run” state). Use the configuration software to verify the configuration and start the main module.
The FX0-GDEV is not supplying any PDO data. PWR ● Green NS ● Green MS ○ OFF	Supply voltage too low	Check the supply voltage.
The FX0-GDEV is not supplying any data. PWR ● Red NS ● Red MS ● Red	Brief supply voltage drop	Check the supply voltage. Reset the Flexi Soft system.
The FX0-GDEV is not supplying any data. PWR ● Green NS ☼ Green (1 Hz) MS ☼ Green (1 Hz)	Wrong node ID or DeviceNet address Incorrect data transmission rate The FX0-GDEV is in the “Idle” state.	Check and correct the node ID and DeviceNet address. Check and correct the data transmission rate.
The FX0-GDEV is not supplying any data. PWR ● Green NS ● Red MS ☼ Red/green	The data transmission rate is incorrect and the FX0-GDEV transceiver is in the “Busoff” state (hardware problem on the physical DeviceNet layer). The cable has been interrupted.	Check and correct the data transmission rate. Check the cabling. Reset the Flexi Soft system.
The FX0-GDEV is not supplying any data. PWR ● Green NS ☼ Green (1 Hz) MS ● Green	The DeviceNet master is in the “Stopped” or “Pre-operational” state. Unable to initialize another slave during bus system initialization. The DeviceNet status of the FX0-GDEV is “Pre-operational”. Wrong node ID or DeviceNet address	Start the DeviceNet master (switch to the “Operational” DeviceNet state). Check that all the slaves on the bus are switched on. Check the cabling. Check whether the DeviceNet master starts automatically. Check and correct the DeviceNet address.
The FX0-GDEV is not supplying any data. PWR ● Green NS ● Red MS ● Green	The FX0-GDEV transceiver is in the “Passive error” state. The cable has been interrupted.	Check the cabling. Use the configuration software to check the diagnostic messages. Reset the Flexi Soft system.

Fault	Possible cause	Possible measures
The FX0-GDEV is not supplying any data. PWR ● Green NS ☼ Red (1 Hz) MS ☼ Red/green	Node guarding or heartbeart consumer failure The guarding configuration has been changed.	Check the cabling. Check the life guarding time (life time factor ≥ 1). Check the heartbeat consumer time (should be ≥ 1.5 × heartbeat producer time). Use the configuration software to check the diagnostic messages. Reset the Flexi Soft system.
The FX0-GDEV is in the “Serious error” state. PWR ● Green NS ● Red MS ☼ Red (2 Hz)	Internal device error on the FX0-GDEV No Flexi Soft gateways are supported by the firmware version of the main module.	Switch the voltage supply for the Flexi Soft system off and then back on again. Use the configuration software to check the diagnostic messages. Use a main module with the required firmware version. If the fault persists, replace the gateway.
The FX0-GDEV/ Flexi Soft system is in the “Serious error” state. PWR ● Red NS ○ OFF MS ● Red	The FX0-GDEV has not been connected correctly to the other Flexi Soft modules. The module connector is contaminated or damaged. There is an internal serious error on another Flexi Soft module.	Plug in the FX0-GDEV correctly. Clean the male and female connectors. Switch the voltage supply back on. Check the other Flexi Soft modules.

7 List of abbreviations

ACR

Automatic Configuration Recovery = a function that allows automatic recovery or duplication of the configuration for connected EFI-enabled safety sensors such as laser scanners or light curtains

CIP

Common Industrial Protocol

COB-ID

Communication Object Identifier = address of the communication object

CoLa

Command Language = SICK-specific configuration and diagnostic protocol

COS

Change Of State = e.g., of a process image

CSV

Comma Separated Values

EDS

Electronic Data Sheet

EFI

Enhanced Function Interface = safe SICK device communication

EIP

EtherNet/IP™ = CIP over Ethernet

EoE

Ethernet over EtherCAT

h

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

INT

Integer = 2 bytes = 1 word

Node ID

Node identifier

OUNID

Originator Unique Node Identifier

PDO

Process Data Object

RPI

Requested Packet Interval = data transmission rate requested by the target device

SCID

Safety Configuration Identifier

SDO

Service Data Object

SINT

Short integer = 1 byte

SNCT

Safety Network Configuration Tool

SNN

Safety Network Number

PLC

Programmable Logic Controller

TUNID

Target Unique Node Identifier

UDINT

Unsigned double integer = 4 bytes = 2 words

UINT

Unsigned integer = 2 bytes = 1 word

USINT

Unsigned short integer = 1 byte

8 Annex

8.1 TCP/IP socket monitor

The TCP/IP Socket Monitor can be used to view the input data sets transferred by a gateway into the network via TCP/IP, to send commands to the gateway, and to write data to the output data sets of the gateway.

The TCP/IP socket monitor is a separate program that is installed with the Flexi Soft Designer. The TCP/IP socket monitor is located in the program directory of the Flexi Soft Designers (the default setting for the installation directory is "C:\Programs\SICK\Flexi-Soft\Tools\TcpIpSocketMonitor").

In order to connect to the gateway via the socket monitor, at least one TCP/IP socket must be enabled on the gateway (see "Ethernet TCP/IP socket interface", page 74).

Using the TCP/IP Socket Monitor

- ► Switch on the Flexi Soft system.
- ► Connect one of the two gateway Ethernet connections to the Ethernet network using a shielded Ethernet cable.
- ► Connect a PC with the same Ethernet network. Make sure that the IP address settings of the computer match the network configuration.

NOTE

The computer can also be directly connected to one of the two gateway Ethernet connections. In this case, either the computer IP address settings or those of the gateway must be adjusted to match the settings of the other device.

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

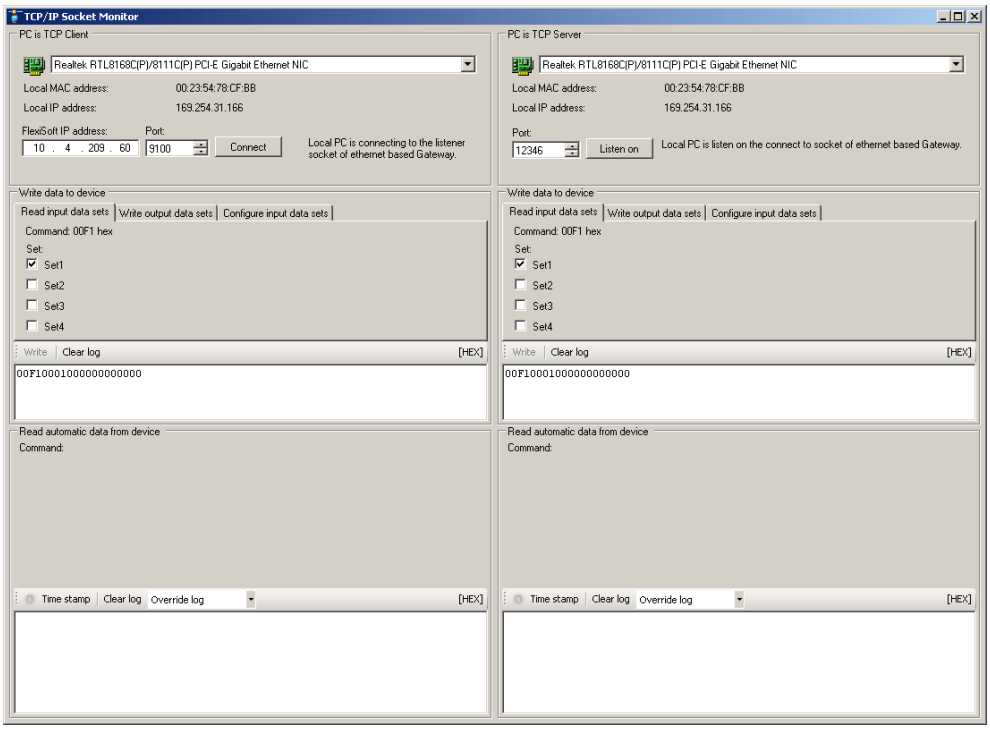

Figure 47: TCP/IP socket monitor

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 as TCP client

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

All available network adapters are displayed in dropdown list in the **PC is TCP Client** area:

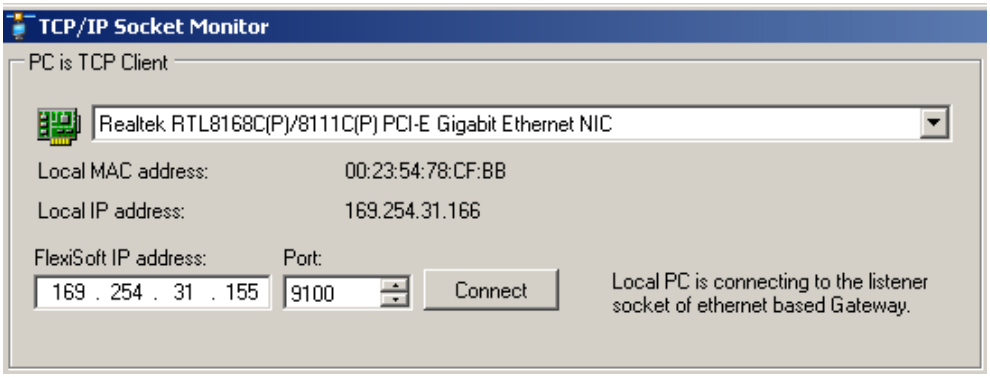

Figure 48: PC is TCP Client area – Gateway is configured to listen

- ► 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 IP address of the gateway under **Flexi Soft IP address** and the **Port** of the activated socket.
- ► Click on **Connect** to establish the connection to the gateway.

PC as 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.

All available network adapters are displayed in dropdown list in the **PC is TCP Server** area:

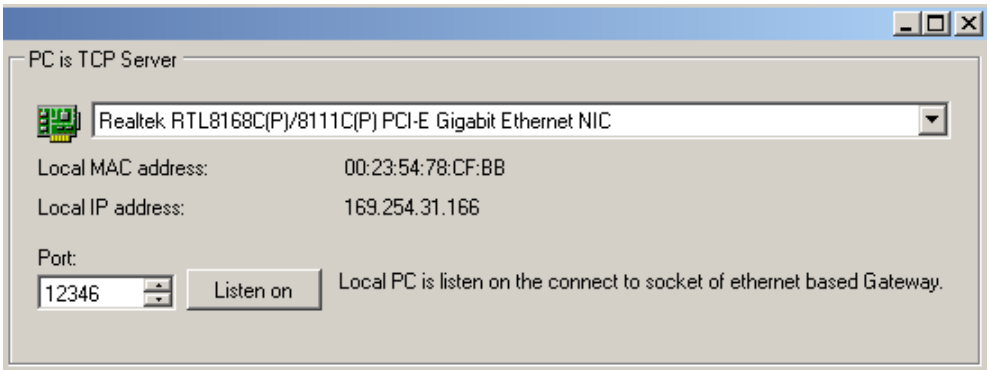

Figure 49: PC is TCP Server area – Gateway is configured to connect

- ► 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** with which the activated socket will establish a connection.
- ► Click on **Listen on** to establish the connection to the gateway.

Once the connection to the gateway is established, the remaining steps are identical irrespective of whether the computer is connected as client or server. It is also possible to establish a client connection and a server connection simultaneously if this is permitted by the gateway configuration.

Write data to device – How to control the gateway

The **Write data to device** area consists of three tabs which can be used to assemble different commands and send them to the gateway. The commands are automatically assembled and displayed in the log window by activating or deactivating the different options on the tabs. Clicking 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 directly in the log window.

On the **Read input data sets** tab, you can request input data from the gateway using the 00F1h command ("figure X").

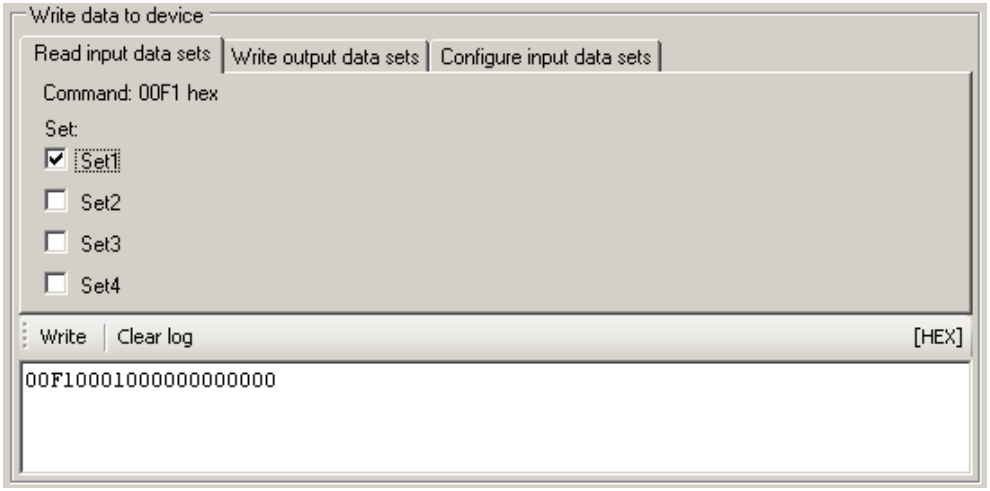

Figure 50: Write data to device – Read input data sets

- ► Activate all data sets that are to be requested. The command that will be sent is shown in the log window.
- ► Click on the **Write** button above the log window to send the displayed command to the gateway. The gateway will respond with a 001Fh message ("figure X") that will be shown in the **Read automatic data from device** area below (see figure 53, page 164).

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 on the **Configure input data sets** tab (see below).

On the **Write output data sets** tab, you can write data to the output data sets using the 00F2h command ("figure X").

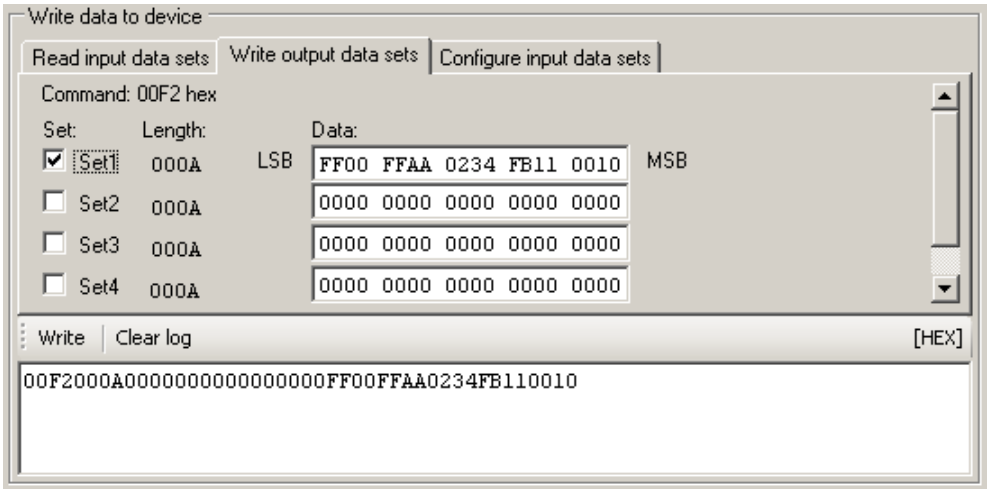

Figure 51: Write data to device – Write output data sets

- ► 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 displayed command to the gateway. The gateway will respond with a 002Fh message to indicate whether the output data sets have been written correctly or if an error has occurred ("figure X"). The result will be shown in the **Read automatic data from device** below (see figure 53, page 164).

On the **Configure input data sets** tab, you can configure the gateway by means of the 00E1h command to send data sets either on change of state (COS) or cyclically (auto update) ("figure X").

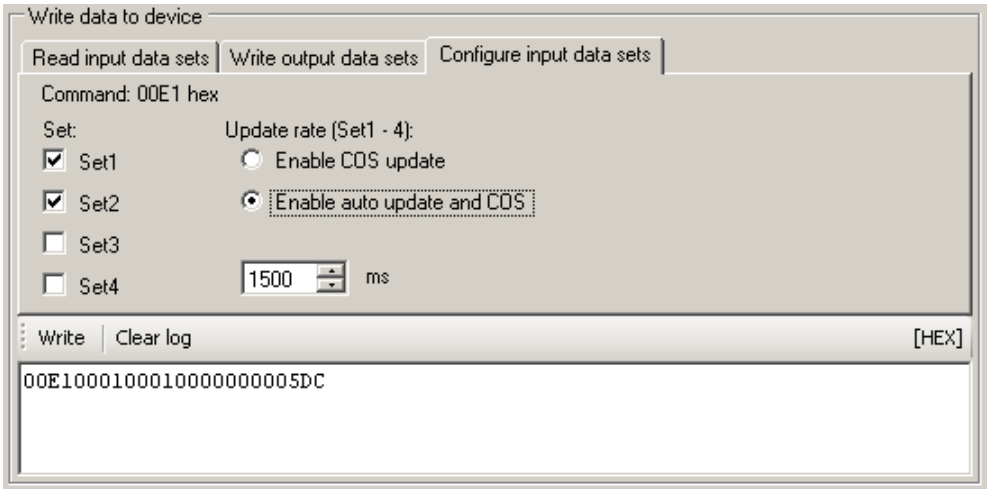

Figure 52: Write data to device – Configure input data sets

- ► Activate all data sets that you want to receive, select 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 the **Write** button above the log window to send the displayed command to the gateway. The gateway will respond with a 001Eh message ("figure X") that will be shown in the **Read automatic data from device** area below (see figure 53, page 164).
- ► Depending on the configuration, the gateway will then follow up with 002Eh messages, i.e. it will send all activated data sets as configured ("figure X").

NOTE

- The initial behavior 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 on the **Configure input data sets** tab.
- The changes you make to the configuration of the input data sets here will not be stored in the Flexi Soft project but will change the behavior 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 messages

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

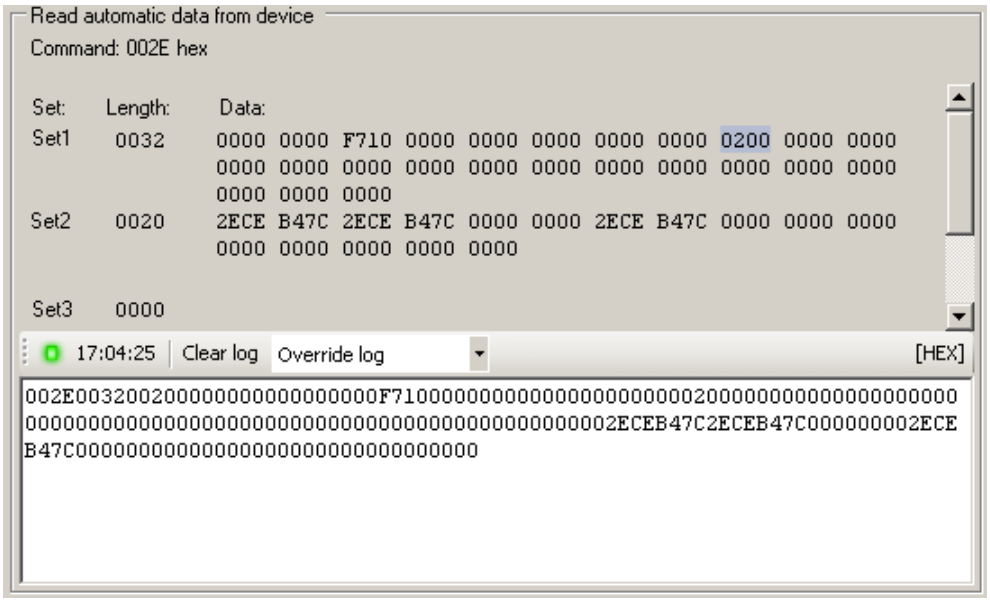

Figure 53: Read automatic data from device

- The last sent **Command** and the **Data Sets**, that have been sent by the gateway are displayed in the gray part of the area. The data is displayed in blocks of four bytes in each case. Data blocks that have changed value since the last message are highlighted blue.
- The log window shows the complete command string that the gateway has sent.
- The communication indicator at the top left corner of the log window title bar lights up green if data is currently being received from the gateway.
- The time stamp in the log window title bar indicates when the last command string was received.
- The **Clear log** button clears the log window.
- You can choose in a dropdown list when the next received command string should overwrite the last one or be appended at the top of the log window.

9 List of figures

10 List of tables

8014526/1KOF/2023-09-08/en

Australia
Phone +61 (3) 9457 0600
1800 33 48 02 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 (0) 2236 62288-0
E-Mail office@sick.at

Belgium/Luxembourg
Phone +32 (0) 2 466 55 66
E-Mail info@sick.be

Brazil
Phone +55 11 3215-4900
E-Mail comercial@sick.com.br

Canada
Phone +1 905.771.1444
E-Mail cs.canada@sick.com

Czech Republic
Phone +420 234 719 500
E-Mail sick@sick.cz

Chile
Phone +56 (2) 2274 7430
E-Mail chile@sick.com

China
Phone +86 20 2882 3600
E-Mail info.china@sick.net.cn

Denmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Finland
Phone +358-9-25 15 800
E-Mail sick@sick.fi

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Germany
Phone +49 (0) 2 11 53 010
E-Mail info@sick.de

Greece
Phone +30 210 6825100
E-Mail office@sick.com.gr

Hong Kong
Phone +852 2153 6300
E-Mail ghk@sick.com.hk

Hungary
Phone +36 1 371 2680
E-Mail ertekesites@sick.hu

India
Phone +91-22-6119 8900
E-Mail info@sick-india.com

Israel
Phone +972 97110 11
E-Mail info@sick-sensors.com

Italy
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 3 5309 2112
E-Mail support@sick.jp

Malaysia
Phone +603-8080 7425
E-Mail enquiry.my@sick.com

Mexico
Phone +52 (472) 748 9451
E-Mail mexico@sick.com

Netherlands
Phone +31 (0) 30 204 40 00
E-Mail info@sick.nl

New Zealand
Phone +64 9 415 0459
0800 222 278 – tollfree
E-Mail sales@sick.co.nz

Norway
Phone +47 67 81 50 00
E-Mail sick@sick.no

Poland
Phone +48 22 539 41 00
E-Mail info@sick.pl

Romania
Phone +40 356-17 11 20
E-Mail office@sick.ro

Singapore
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Slovakia
Phone +421 482 901 201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 591 78849
E-Mail office@sick.si

South Africa
Phone +27 10 060 0550
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321/4
E-Mail infokorea@sick.com

Spain
Phone +34 93 480 31 00
E-Mail info@sick.es

Sweden
Phone +46 10 110 10 00
E-Mail info@sick.se

Switzerland
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Taiwan
Phone +886-2-2375-6288
E-Mail sales@sick.com.tw

Thailand
Phone +66 2 645 0009
E-Mail marcom.th@sick.com

Turkey
Phone +90 (216) 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 (0) 4 88 65 878
E-Mail contact@sick.ae

United Kingdom
Phone +44 (0)17278 31121
E-Mail info@sick.co.uk

USA
Phone +1 800.325.7425
E-Mail info@sick.com

Vietnam
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Detailed addresses and further locations at **www.sick.com**

SICK AG | Waldkirch | Germany | www.sick.com

SICK
Sensor Intelligence.